

Языки программирования

Контекстно-свободные грамматики

Содержание

- Понятие контекстно-свободной грамматики
- КС-грамматики и конечные распознаватели
- Синтаксически управляемые процессы обработки языков
- Атрибутные транслирующие грамматики

Формальные грамматики

- *Формальный язык* – множество символьных цепочек.
- *Формальная грамматика* – набор правил, с помощью которых порождаются цепочки формального языка.
- Формальные грамматики можно преобразовать в конечные распознаватели и обрабатывающие автоматы, которые распознают/транслируют соответствующие множества цепочек.

Пример формальной грамматики

1. $\langle \text{предложение} \rangle \rightarrow \langle \text{подлежащее} \rangle \langle \text{сказуемое} \rangle \langle \text{дополнение} \rangle$
2. $\langle \text{подлежащее} \rangle \rightarrow \langle \text{прилагательное} \rangle \langle \text{существительное} \rangle$
3. $\langle \text{дополнение} \rangle \rightarrow \langle \text{прилагательное} \rangle \langle \text{существительное} \rangle$
4. $\langle \text{сказуемое} \rangle \rightarrow \text{сдаёт}$
5. $\langle \text{прилагательное} \rangle \rightarrow \text{каждый}$
6. $\langle \text{существительное} \rangle \rightarrow \text{экзамен}$
7. $\langle \text{существительное} \rangle \rightarrow \text{студент}$

Пример формальной грамматики



Контекстно-свободная грамматика

- Для задания *КС-грамматики* необходимо:
 - конечное множество *терминалов* – символов, не требующих дополнительных определений
 - конечное множество *нетерминалов* – символов, которые определяются через терминалы и другие нетерминалы
 - конечное множество *правил (продукций)* – определений вида $\langle A \rangle \rightarrow \alpha$, где
 - левая часть* $\langle A \rangle$ – нетерминал
 - правая часть* α – конечная, возможно пустая, цепочка терминалов и нетерминалов
 - начальный нетерминал

Подстановки и выводы

- *Подстановка* – замена нетерминала правой частью определяющей его продукции.
- *Вывод* – последовательность подстановок.
- *КС-язык* – множество терминальных цепочек, которые можно вывести из начального нетерминала КС-грамматики.

Пример вывода

$$1. \langle S \rangle \rightarrow a \langle A \rangle \langle B \rangle c$$

$$2. \langle S \rangle \rightarrow \varepsilon$$

$$3. \langle A \rangle \rightarrow c \langle S \rangle \langle B \rangle$$

$$4. \langle A \rangle \rightarrow \langle A \rangle b$$

$$5. \langle B \rangle \rightarrow b \langle B \rangle$$

$$6. \langle B \rangle \rightarrow a$$

$$\bullet \langle S \rangle_1 \Rightarrow a \langle A \rangle \langle B \rangle c$$

$$\bullet a \langle A \rangle_4 \langle B \rangle c \Rightarrow a \langle A \rangle b \langle B \rangle c$$

$$\bullet a \langle A \rangle_3 b \langle B \rangle c \Rightarrow a c \langle S \rangle \langle B \rangle b \langle B \rangle c$$

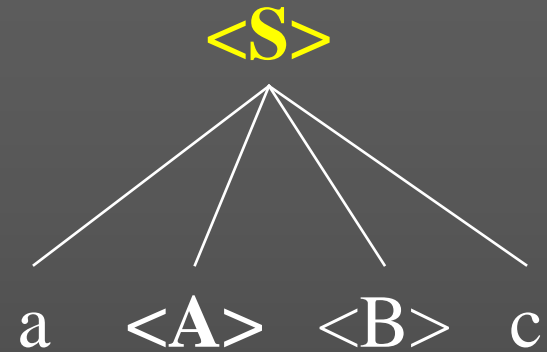
$$\bullet a c \langle S \rangle \langle B \rangle_6 b \langle B \rangle c \Rightarrow a c \langle S \rangle a b \langle B \rangle c$$

$$\bullet a c \langle S \rangle_2 a b \langle B \rangle c \Rightarrow a c a b \langle B \rangle c$$

$$\bullet a c a b \langle B \rangle_6 c \Rightarrow a c a b a c, \text{ то есть } \langle S \rangle \Rightarrow^* a c a b a c$$

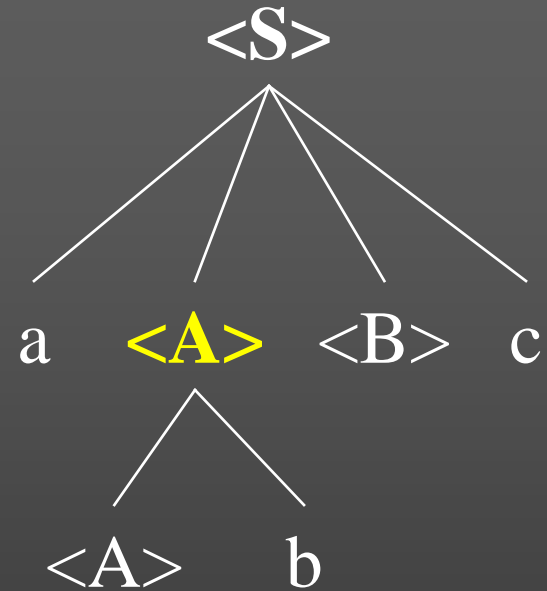
Пример вывода

- $\langle S \rangle_1 \Rightarrow a \langle A \rangle \langle B \rangle c$



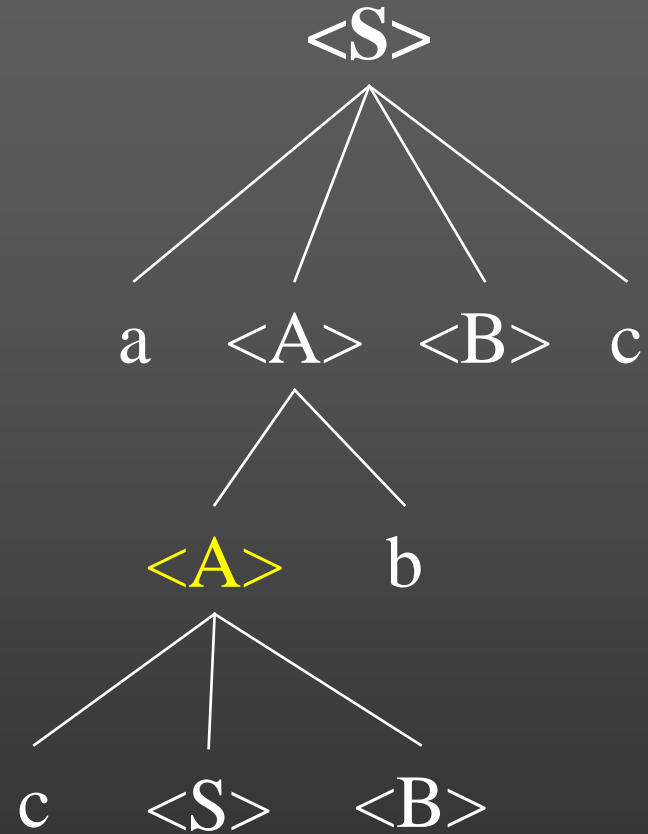
Пример вывода

- $\langle S \rangle_1 \Rightarrow a \langle A \rangle \langle B \rangle c$
- $a \langle A \rangle_4 \langle B \rangle c \Rightarrow a \langle A \rangle b \langle B \rangle c$



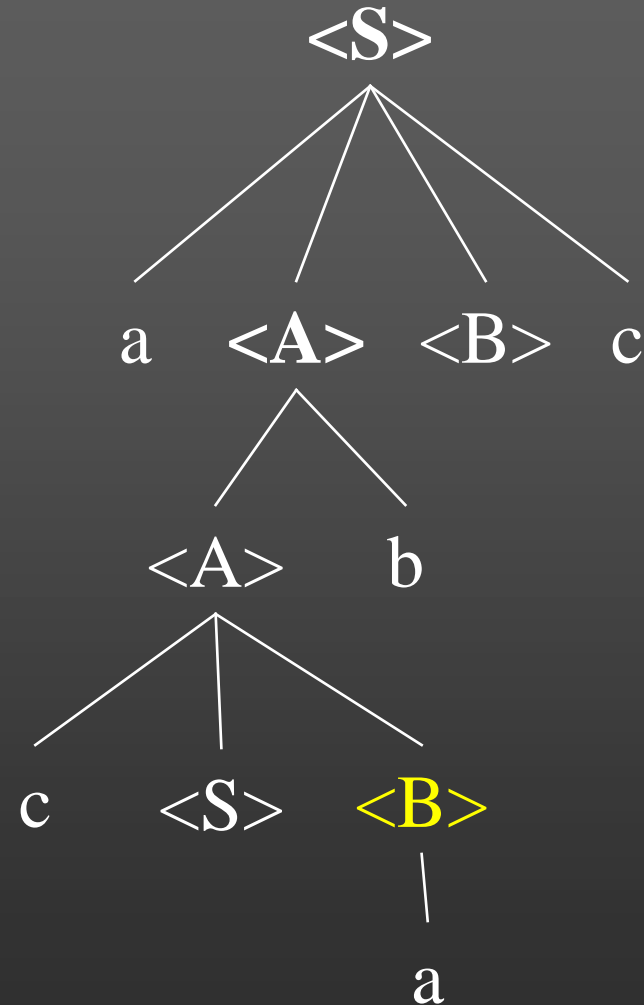
Пример вывода

- $\langle S \rangle_1 \Rightarrow a \langle A \rangle \langle B \rangle c$
- $a \langle A \rangle_4 \langle B \rangle c \Rightarrow a \langle A \rangle b \langle B \rangle c$
- $a \langle A \rangle_3 b \langle B \rangle c \Rightarrow ac \langle S \rangle \langle B \rangle b \langle B \rangle c$



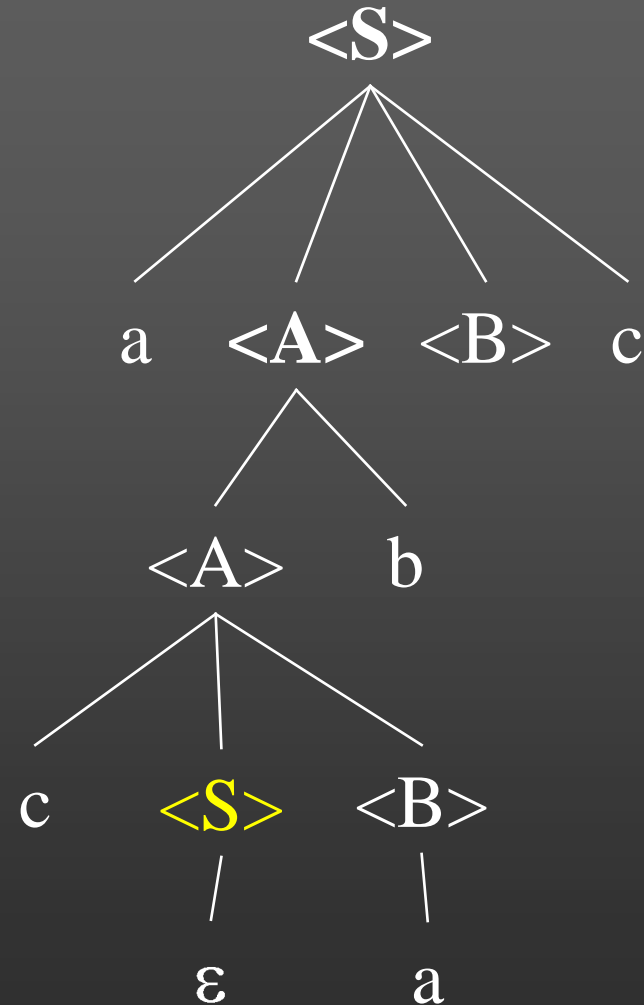
Пример вывода

- $\langle S \rangle_1 \Rightarrow a \langle A \rangle \langle B \rangle c$
- $a \langle A \rangle_4 \langle B \rangle c \Rightarrow a \langle A \rangle b \langle B \rangle c$
- $a \langle A \rangle_3 b \langle B \rangle c \Rightarrow ac \langle S \rangle \langle B \rangle b \langle B \rangle c$
- $ac \langle S \rangle \langle B \rangle_6 b \langle B \rangle c \Rightarrow ac \langle S \rangle ab \langle B \rangle c$



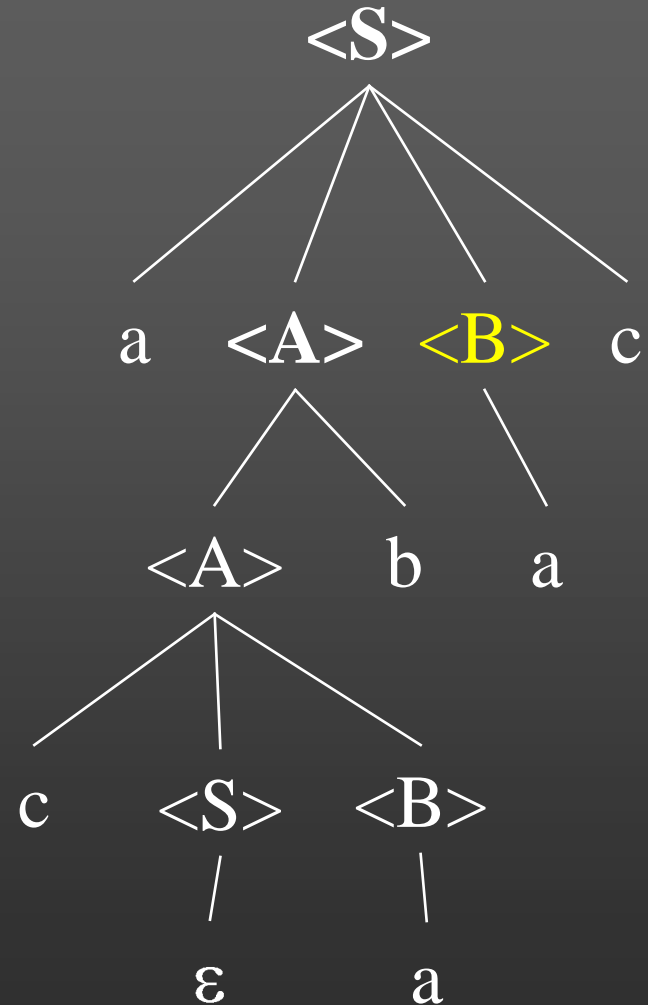
Пример вывода

- $\langle S \rangle_1 \Rightarrow a \langle A \rangle \langle B \rangle c$
- $a \langle A \rangle_4 \langle B \rangle c \Rightarrow a \langle A \rangle b \langle B \rangle c$
- $a \langle A \rangle_3 b \langle B \rangle c \Rightarrow ac \langle S \rangle \langle B \rangle b \langle B \rangle c$
- $ac \langle S \rangle \langle B \rangle_6 b \langle B \rangle c \Rightarrow ac \langle S \rangle ab \langle B \rangle c$
- $ac \langle S \rangle_2 ab \langle B \rangle c \Rightarrow acab \langle B \rangle c$



Пример вывода

- $\langle S \rangle_1 \Rightarrow a \langle A \rangle \langle B \rangle c$
- $a \langle A \rangle_4 \langle B \rangle c \Rightarrow a \langle A \rangle b \langle B \rangle c$
- $a \langle A \rangle_3 b \langle B \rangle c \Rightarrow ac \langle S \rangle \langle B \rangle b \langle B \rangle c$
- $ac \langle S \rangle \langle B \rangle_6 b \langle B \rangle c \Rightarrow ac \langle S \rangle ab \langle B \rangle c$
- $ac \langle S \rangle_2 ab \langle B \rangle c \Rightarrow acab \langle B \rangle c$
- $acab \langle B \rangle_6 c \Rightarrow acabac$



Неоднозначность вывода

- Цепочке языка может соответствовать более чем один вывод.

1. $\langle S \rangle \rightarrow a \langle A \rangle \langle B \rangle c$

2. $\langle S \rangle \rightarrow \varepsilon$

3. $\langle A \rangle \rightarrow c \langle S \rangle \langle B \rangle$

4. $\langle A \rangle \rightarrow \langle A \rangle b$

5. $\langle B \rangle \rightarrow b \langle B \rangle$

6. $\langle B \rangle \rightarrow a$

● $\langle S \rangle_1 \Rightarrow a \langle A \rangle \langle B \rangle c$

● $a \langle A \rangle_4 \langle B \rangle c \Rightarrow a \langle A \rangle b \langle B \rangle c$

● $a \langle A \rangle_3 b \langle B \rangle c \Rightarrow a c \langle S \rangle \langle B \rangle b \langle B \rangle c$

● $a c \langle S \rangle \langle B \rangle_6 b \langle B \rangle c \Rightarrow a c \langle S \rangle a b \langle B \rangle c$

● $a c \langle S \rangle_2 a b \langle B \rangle c \Rightarrow a c a b \langle B \rangle c$

● $a c a b \langle B \rangle_6 c \Rightarrow \mathbf{a c a b a c}$

● $\langle S \rangle_1 \Rightarrow a \langle A \rangle \langle B \rangle c$

● $a \langle A \rangle \langle B \rangle_5 c \Rightarrow a \langle A \rangle b \langle B \rangle c$

● $a \langle A \rangle_3 b \langle B \rangle c \Rightarrow a c \langle S \rangle \langle B \rangle b \langle B \rangle c$

● $a c \langle S \rangle \langle B \rangle_6 b \langle B \rangle c \Rightarrow a c \langle S \rangle a b \langle B \rangle c$

● $a c \langle S \rangle_2 a b \langle B \rangle c \Rightarrow a c a b \langle B \rangle c$

● $a c a b \langle B \rangle_6 c \Rightarrow \mathbf{a c a b a c}$

Неоднозначность КС-грамматики

- Если вывод каждой цепочки КС-языка единственный, то соответствующая КС-грамматика называется *однозначной*, иначе — *неоднозначной*.

Лево- и правосторонние выводы

- *Лево-/правосторонний вывод* – последовательность подстановок, в которой на каждом шаге заменяется самый левый/правый нетерминал.
- Каждому дереву вывода соответствует единственный левосторонний и единственный правосторонний выводы.

Пример левостороннего вывода

1. $\langle S \rangle \rightarrow a \langle A \rangle \langle B \rangle c$

4. $\langle A \rangle \rightarrow \langle A \rangle b$

2. $\langle S \rangle \rightarrow \varepsilon$

5. $\langle B \rangle \rightarrow b \langle B \rangle$

3. $\langle A \rangle \rightarrow c \langle S \rangle \langle B \rangle$

6. $\langle B \rangle \rightarrow a$

• $\langle S \rangle_1 \Rightarrow_L a \langle A \rangle \langle B \rangle c$

• $a \langle A \rangle_4 \langle B \rangle c \Rightarrow_L a \langle A \rangle b \langle B \rangle c$

• $a \langle A \rangle_3 b \langle B \rangle c \Rightarrow_L a c \langle S \rangle \langle B \rangle b \langle B \rangle c$

• $a c \langle S \rangle_2 \langle B \rangle b \langle B \rangle c \Rightarrow_L a c \langle B \rangle b \langle B \rangle c$

• $a c \langle B \rangle_6 b \langle B \rangle c \Rightarrow_L a c a b \langle B \rangle c$

• $a c a b \langle B \rangle_6 c \Rightarrow_L a c a b a c$, то есть $\langle S \rangle \Rightarrow_L^* a c a b a c$

Пример правостороннего вывода

$$1. \langle S \rangle \rightarrow a \langle A \rangle \langle B \rangle c$$

$$2. \langle S \rangle \rightarrow \varepsilon$$

$$3. \langle A \rangle \rightarrow c \langle S \rangle \langle B \rangle$$

$$4. \langle A \rangle \rightarrow \langle A \rangle b$$

$$5. \langle B \rangle \rightarrow b \langle B \rangle$$

$$6. \langle B \rangle \rightarrow a$$

$$\bullet \langle S \rangle_1 \Rightarrow_{\mathbf{R}} a \langle A \rangle \langle B \rangle c$$

$$\bullet a \langle A \rangle \langle B \rangle_6 c \Rightarrow_{\mathbf{R}} a \langle A \rangle a c$$

$$\bullet a \langle A \rangle_4 a c \Rightarrow_{\mathbf{R}} a \langle A \rangle b a c$$

$$\bullet a \langle A \rangle_3 b a c \Rightarrow_{\mathbf{R}} a c \langle S \rangle \langle B \rangle b a c$$

$$\bullet a c \langle S \rangle \langle B \rangle_6 b a c \Rightarrow_{\mathbf{R}} a c \langle S \rangle a b a c$$

$$\bullet a c \langle S \rangle_2 a b a c \Rightarrow_{\mathbf{R}} a c a b a c, \text{ то есть } \langle S \rangle \Rightarrow_{\mathbf{R}}^* a c a b a c$$

Упражнение

- Построить КС-грамматику констант языка Pascal.

<константа> → ?

- Вывести константы 'ABRACADABRA', 123, \$123, 1E23 с помощью лево/правостороннего вывода.

КС-грамматики и конечные распознаватели

- *Утверждение 1.*

Регулярные множества являются КС-языками.

- *Доказательство*

Построение КС-грамматики регулярного множества конечного распознавателя:

1. Терминалы: символы алфавита распознавателя
2. Нетерминалы: множество состояний
3. Правила:
для перехода $A^x \rightarrow B$ продукция $\langle A \rangle \rightarrow x \langle B \rangle$
для допускающего состояния A правило $\langle A \rangle \rightarrow \varepsilon$
4. Начальный нетерминал: начальное состояние.

Пример построения КС-грамматики для конечного распознавателя

- «Контролер нечетности единиц» с регулярным множеством всех цепочек, состоящих из 0 и 1 и имеющих нечетное число единиц.

	0	1
ЧЕТ	ЧЕТ	НЕЧЕТ
НЕЧЕТ	НЕЧЕТ	ЧЕТ

0

1



1. $\langle \text{ЧЕТ} \rangle \rightarrow 1 \langle \text{НЕЧЕТ} \rangle$
2. $\langle \text{ЧЕТ} \rangle \rightarrow 0 \langle \text{ЧЕТ} \rangle$
3. $\langle \text{НЕЧЕТ} \rangle \rightarrow 0 \langle \text{НЕЧЕТ} \rangle$
4. $\langle \text{НЕЧЕТ} \rangle \rightarrow 1 \langle \text{ЧЕТ} \rangle$
5. $\langle \text{НЕЧЕТ} \rangle \rightarrow \varepsilon$

КС-грамматики и конечные распознаватели

- *Утверждение 2.*

Если КС-грамматика содержит только продукции вида $\langle A \rangle \rightarrow x \langle B \rangle$ и $\langle A \rangle \rightarrow \varepsilon$, то существует конечный распознаватель для соответствующего КС-языка.

- *Доказательство*

Построение конечного распознавателя:

1. Алфавит: множество терминалов
2. Состояния: множество нетерминалов
3. Переходы: для правила $\langle A \rangle \rightarrow x \langle B \rangle$ переход $A^x \rightarrow B$
4. Допускающие состояния: A в переходе $\langle A \rangle \rightarrow \varepsilon$
5. Начальное состояние: начальный нетерминал.

Пример построения конечного распознавателя для КС-грамматики

- КС-грамматика идентификатора Pascal
 - 1.<идентификатор> \rightarrow L<цепочка букв и цифр>
 - 2.<цепочка букв и цифр> \rightarrow L<цепочка букв и цифр>
 - 3.<цепочка букв и цифр> \rightarrow D<цепочка букв и цифр>
 - 4.<цепочка букв и цифр> \rightarrow ϵ
- Конечный распознаватель идентификатора Pascal (недетерминированный!):

	<i>L</i>	<i>D</i>	
\rightarrow идентификатор	цепочка букв и цифр		0
цепочка букв и цифр	цепочка букв и цифр	цепочка букв и цифр	1

Праволинейные КС-грамматики

- КС-грамматика называется *праволинейной*, если ее продукции имеют вид $\langle A \rangle \rightarrow w \langle B \rangle$ или $\langle A \rangle \rightarrow w$, где $\langle A \rangle$ и $\langle B \rangle$ – нетерминалы, w – цепочка терминалов (возможно пустая).
- Пример праволинейной грамматики:
 $\langle \text{описание} \rangle \rightarrow \langle \text{тип} \rangle \langle \text{идентификатор} \rangle \langle \text{список переменных} \rangle ;$
 $\langle \text{список переменных} \rangle \rightarrow , \langle \text{идентификатор} \rangle \langle \text{список переменных} \rangle$
 $\langle \text{список переменных} \rangle \rightarrow \varepsilon$
int a, b, c, i, j, k;
- Праволинейные грамматики можно преобразовать в грамматики специального вида (из Утверждения 2).

Пример преобразования праволинейной КС-грамматики

1. $\langle S \rangle \rightarrow a\langle A \rangle$

2. $\langle S \rangle \rightarrow bc$

3. $\langle S \rangle \rightarrow \langle A \rangle$

4. $\langle A \rangle \rightarrow abb\langle S \rangle$

5. $\langle A \rangle \rightarrow c\langle A \rangle$

6. $\langle A \rangle \rightarrow \varepsilon$

1. $\langle S \rangle \rightarrow a\langle A \rangle$

2. $\langle S \rangle \rightarrow b\langle cEpsilon \rangle$

3. $\langle cEpsilon \rangle \rightarrow c\langle Epsilon \rangle$

4. $\langle Epsilon \rangle \rightarrow \varepsilon$

5. $\langle S \rangle \rightarrow a\langle bbS \rangle$

6. $\langle S \rangle \rightarrow c\langle A \rangle$

7. $\langle S \rangle \rightarrow \varepsilon$

8. $\langle A \rangle \rightarrow a\langle bbS \rangle$

9. $\langle bbS \rangle \rightarrow b\langle bS \rangle$

10. $\langle bS \rangle \rightarrow b\langle S \rangle$

11. $\langle A \rangle \rightarrow c\langle A \rangle$

12. $\langle A \rangle \rightarrow \varepsilon$

Пример преобразования праволинейной КС-грамматики

1. $\langle S \rangle \rightarrow a \langle A \rangle$
2. $\langle S \rangle \rightarrow b \langle c \text{Epsilon} \rangle$
3. $\langle c \text{Epsilon} \rangle \rightarrow c \langle \text{Epsilon} \rangle$
4. $\langle \text{Epsilon} \rangle \rightarrow \varepsilon$
5. $\langle S \rangle \rightarrow a \langle bbS \rangle$
6. $\langle S \rangle \rightarrow c \langle A \rangle$
7. $\langle S \rangle \rightarrow \varepsilon$
8. $\langle A \rangle \rightarrow a \langle bbS \rangle$
9. $\langle bbS \rangle \rightarrow b \langle bS \rangle$
10. $\langle bS \rangle \rightarrow b \langle S \rangle$
11. $\langle A \rangle \rightarrow c \langle A \rangle$
12. $\langle A \rangle \rightarrow \varepsilon$

$\rightarrow S$

$c \text{Epsilon}$

Epsilon

A

bbS

bS

	a	b	c	
$\rightarrow S$	A, bbS	cEpsilon	A	1
$c \text{Epsilon}$			Epsilon	0
Epsilon				1
A	bbS		A	1
bbS		bS		0
bS		S		0

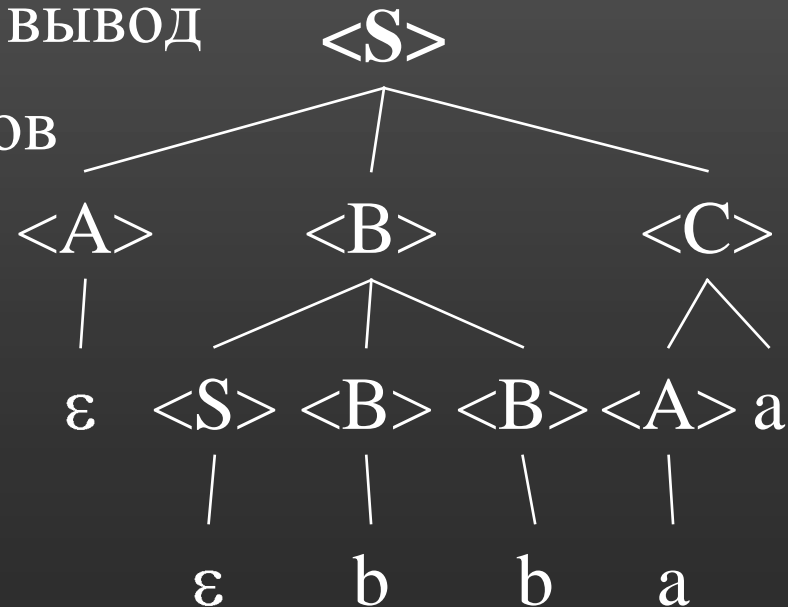
Упражнения

1. Написать алгоритм преобразования произвольной праволинейной грамматики в грамматику специального вида.
2. КС-грамматика порождает дерево вывода:

a) построить левосторонний вывод

b) найти общее число выводов

c) найти дерево вывода для цепочки ab



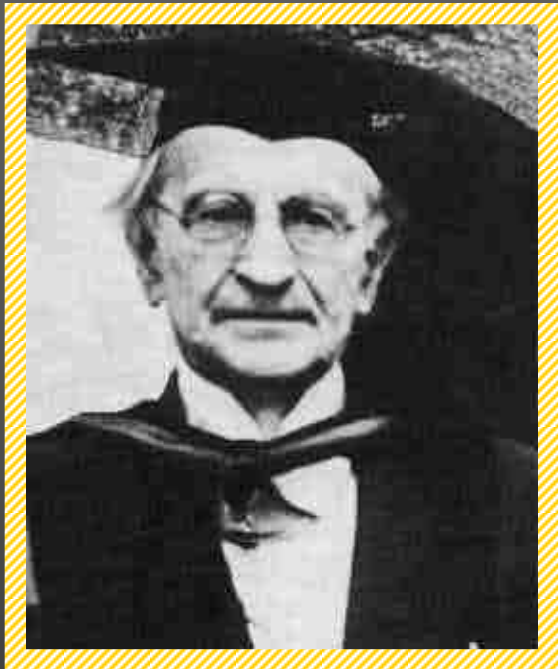
Синтаксически управляемые процессы обработки языков

- *Синтаксически управляемая* обработка КС-языка основана на обработке каждого отдельного правила соответствующей КС-грамматики.
- Синтаксически управляемые процессы используются в решении задачи распознавания КС-языков.

Транслирующие грамматики

- *Транслирующая грамматика (грамматика перевода)* – КС-грамматика, терминалы которой разбиты на два подмножества:
 - *входные символы* грамматики;
 - *символы действия* – подпрограммы обработки (преобразования, печати и др.) входных символов.
- Цепочка языка, определяемого транслирующей грамматикой, называется *последовательностью актов*.
- Рассмотрим задачу построения грамматики перевода арифметических выражений в польскую запись.

Польская запись



Ян Лукашевич
1878 – 1956 гг.

- Обычная форма записи арифметических выражений – *инфиксная*.

$$a+b*c$$

$$a*b+c$$

- В *постфиксной (польской) записи* знак операции помещается *после* операндов.

$$abc*+$$

$$ab*c+$$

Польская запись

- Польская запись никогда не содержит скобок:
 $(a+b)*c \Rightarrow ab+c*$
 $a+b*(c+d)*(e+f) \Rightarrow abcd+*ef+*+$
- Выражения в польской записи легко вычислять.
- В структуру синтаксического блока некоторых компиляторов включается *специальный блок перевода арифметических выражений в постфиксную форму.*

Вычисление значения выражения в польской записи

```
STACK.INIT;  
while not (конец_цепочки) do  
  case тип_текущего_символа of  
    операнд      :  
      STACK.PUSH(значение_операнда);  
    знак_операции :  
      begin  
        arg1 := STACK.POP;  
        arg2 := STACK.POP;  
        STACK.PUSH(arg1 знак_операции arg2);  
      end;  
  end;  
end;  
Результат := STACK.TOP;
```

КС-грамматика польской записи

1. $\langle \text{операнд} \rangle \rightarrow \langle \text{операнд} \rangle \langle \text{операнд} \rangle \langle \text{знак} \rangle$
2. $\langle \text{операнд} \rangle \rightarrow I$
3. $\langle \text{знак} \rangle \rightarrow +$
4. $\langle \text{знак} \rangle \rightarrow *$
5. $\langle \text{знак} \rangle \rightarrow /$
6. $\langle \text{знак} \rangle \rightarrow \dots$

где

I – идентификатор

Перевод в польскую запись

- $a+b*c \Rightarrow abc*+$
- Шаги перевода:
 - `read(a); write(a);`
 - `read(+); read(b);`
 - `write(b); read(*);`
 - `read(c); write(c);`
 - `write(*); write(+);`
- Последовательность актов:
 $a\{a\}+b\{b\}*c\{c\}\{*\}\{+\}$

Грамматика перевода инфиксных выражений

$$1. \langle E \rangle \rightarrow \langle E \rangle + \langle T \rangle$$

$$2. \langle E \rangle \rightarrow \langle T \rangle$$

$$3. \langle T \rangle \rightarrow \langle T \rangle * \langle P \rangle$$

$$4. \langle T \rangle \rightarrow \langle P \rangle$$

$$5. \langle P \rangle \rightarrow (\langle E \rangle)$$

$$6. \langle P \rangle \rightarrow a$$

$$7. \langle P \rangle \rightarrow b$$

$$8. \langle P \rangle \rightarrow c$$



$$1. \langle E \rangle \rightarrow \langle E \rangle + \langle T \rangle \{+\}$$

$$2. \langle E \rangle \rightarrow \langle T \rangle$$

$$3. \langle T \rangle \rightarrow \langle T \rangle * \langle P \rangle \{*\}$$

$$4. \langle T \rangle \rightarrow \langle P \rangle$$

$$5. \langle P \rangle \rightarrow (\langle E \rangle)$$

$$6. \langle P \rangle \rightarrow a \{a\}$$

$$7. \langle P \rangle \rightarrow b \{b\}$$

$$8. \langle P \rangle \rightarrow c \{c\}$$

Перевод инфиксных выражений

$$1. \langle E \rangle \rightarrow \langle E \rangle + \langle T \rangle \{ + \}$$

$$2. \langle E \rangle \rightarrow \langle T \rangle$$

$$3. \langle T \rangle \rightarrow \langle T \rangle * \langle P \rangle \{ * \}$$

$$4. \langle T \rangle \rightarrow \langle P \rangle$$

$$5. \langle P \rangle \rightarrow (\langle E \rangle)$$

$$6. \langle P \rangle \rightarrow a \{ a \}$$

$$7. \langle P \rangle \rightarrow b \{ b \}$$

$$8. \langle P \rangle \rightarrow c \{ c \}$$

- $\langle E \rangle \Rightarrow_L (a+b)*c$

$$\langle E \rangle_2 \Rightarrow \langle T \rangle_3 \Rightarrow \langle T \rangle_4 * \langle P \rangle \{ * \} \Rightarrow \langle P \rangle_5 * \langle P \rangle \{ * \} \Rightarrow$$

$$(\langle E \rangle_1) * \langle P \rangle \{ * \} \Rightarrow (\langle E \rangle + \langle T \rangle \{ + \}) * \langle P \rangle \{ * \} \Rightarrow *$$

$$(a \{ a \} + b \{ b \} \{ + \}) * c \{ c \} \{ * \}$$

$$\{ a \} \{ b \} \{ + \} \{ c \} \{ * \} \equiv ab+c*$$

Синтаксически управляемый перевод

- *Входная грамматика* – транслирующая грамматика с вычеркнутыми символами действия.
Входной язык – язык входной грамматики.
- *Выходная грамматика* – транслирующая грамматика с вычеркнутыми входными символами.
Выходной язык – язык выходной грамматики.
- *Транслирующая грамматика* – грамматика перевода цепочек входного языка в цепочки выходного языка.
- *Перевод, определяемый транслирующей грамматикой*, – множество пар (подпоследовательность входных символов, подпоследовательность действий).
- *Замечание*. Перевод можно определить более чем одной транслирующей грамматикой.

Упражнения

1. Вычислить постфиксные выражения:

- $3\ 7\ 8\ +\ * \ 4\ -$
- $6\ 9\ 5\ 2\ * \ + \ *$
- $1\ 2\ 3\ -\ 4\ 5\ * \ 6\ 7\ + \ * \ + \ -$

2. Перевести в постфиксную форму:

- $a+b*c*(a+b)*(a+c)$
- $a+(b+c)*((a+b)*c+a)$

3. Построить грамматику перевода инфиксных выражений в функциональные. Например:

$$a+b \Rightarrow \text{PLUS}(a,b)$$

$$a*b \Rightarrow \text{MUL}(a,b)$$

$$a+b*c \Rightarrow \text{PLUS}(a, \text{MUL}(b,c))$$

Содержание

- Понятие контекстно-свободной грамматики
- КС-грамматики и конечные распознаватели
- Синтаксически управляемые процессы обработки языков
- **Атрибутные транслирующие грамматики**

Атрибутные транслирующие грамматики

- *Атрибутные транслирующие грамматики* позволяют не только переводить цепочки, но и вычислять их значение в некотором смысле.
- Входные символы, символы действия и нетерминалы атрибутной транслирующей грамматики обладают конечным множеством *атрибутов* – свойств, которые используются для вычисления значения цепочки.
- Атрибуты делятся на *синтезируемые* и *наследуемые*.

Пример синтезируемых атрибутов

- Как построить синтаксический блок, вычисляющий арифметические выражения, состоящие из символов $() + * C$?
 - Данные символы являются лексемами, полученными от лексического блока;
 - C – константа с некоторым значением.
- Используем следующую грамматику перевода:

1. $\langle S \rangle \rightarrow \langle E \rangle \{\text{ОТВЕТ}\}$

5. $\langle T \rangle \rightarrow \langle P \rangle$

2. $\langle E \rangle \rightarrow \langle E \rangle + \langle T \rangle$

6. $\langle P \rangle \rightarrow (\langle E \rangle)$

3. $\langle E \rangle \rightarrow \langle T \rangle$

7. $\langle P \rangle \rightarrow C$

4. $\langle T \rangle \rightarrow \langle T \rangle * \langle P \rangle$

Пример синтезируемых атрибутов

$$1. \langle S \rangle \rightarrow \langle E \rangle \{ \text{ОТВЕТ} \}$$

$$5. \langle T \rangle \rightarrow \langle P \rangle$$

$$2. \langle E \rangle \rightarrow \langle E \rangle + \langle T \rangle$$

$$6. \langle P \rangle \rightarrow (\langle E \rangle)$$

$$3. \langle E \rangle \rightarrow \langle T \rangle$$

$$7. \langle P \rangle \rightarrow C$$

$$4. \langle T \rangle \rightarrow \langle T \rangle * \langle P \rangle$$

- Рассмотрим цепочку $S = (C_3 + C_8) * (C_1 + C_9)$, где индексы – *атрибуты*, обозначающие значение константы.
- Как осуществить перевод $S \Rightarrow \text{ОТВЕТ}_{110}$?
- Очевидное решение – сопоставить каждому нетерминалу значение атрибута в порождаемом им выражении.

Пример синтезируемых атрибутов

1. $\langle S \rangle \rightarrow \langle E \rangle \{ \text{ОТВЕТ} \}$

2. $\langle E \rangle \rightarrow \langle E \rangle + \langle T \rangle$

3. $\langle E \rangle \rightarrow \langle T \rangle$

4. $\langle T \rangle \rightarrow \langle T \rangle * \langle P \rangle$

5. $\langle T \rangle \rightarrow \langle P \rangle$

6. $\langle P \rangle \rightarrow (\langle E \rangle)$

7. $\langle P \rangle \rightarrow C$



1. $\langle S \rangle \rightarrow \langle E \rangle_q \{ \text{ОТВЕТ}_r \}$

$r \leftarrow q$

2. $\langle E \rangle_p \rightarrow \langle E \rangle_q + \langle T \rangle_r$

$p \leftarrow q + r$

3. $\langle E \rangle_p \rightarrow \langle T \rangle_q$

$p \leftarrow q$

4. $\langle T \rangle_p \rightarrow \langle T \rangle_q * \langle P \rangle_r$

$p \leftarrow q * r$

5. $\langle T \rangle_p \rightarrow \langle P \rangle_q$

$p \leftarrow q$

6. $\langle P \rangle_p \rightarrow (\langle E \rangle)_q$

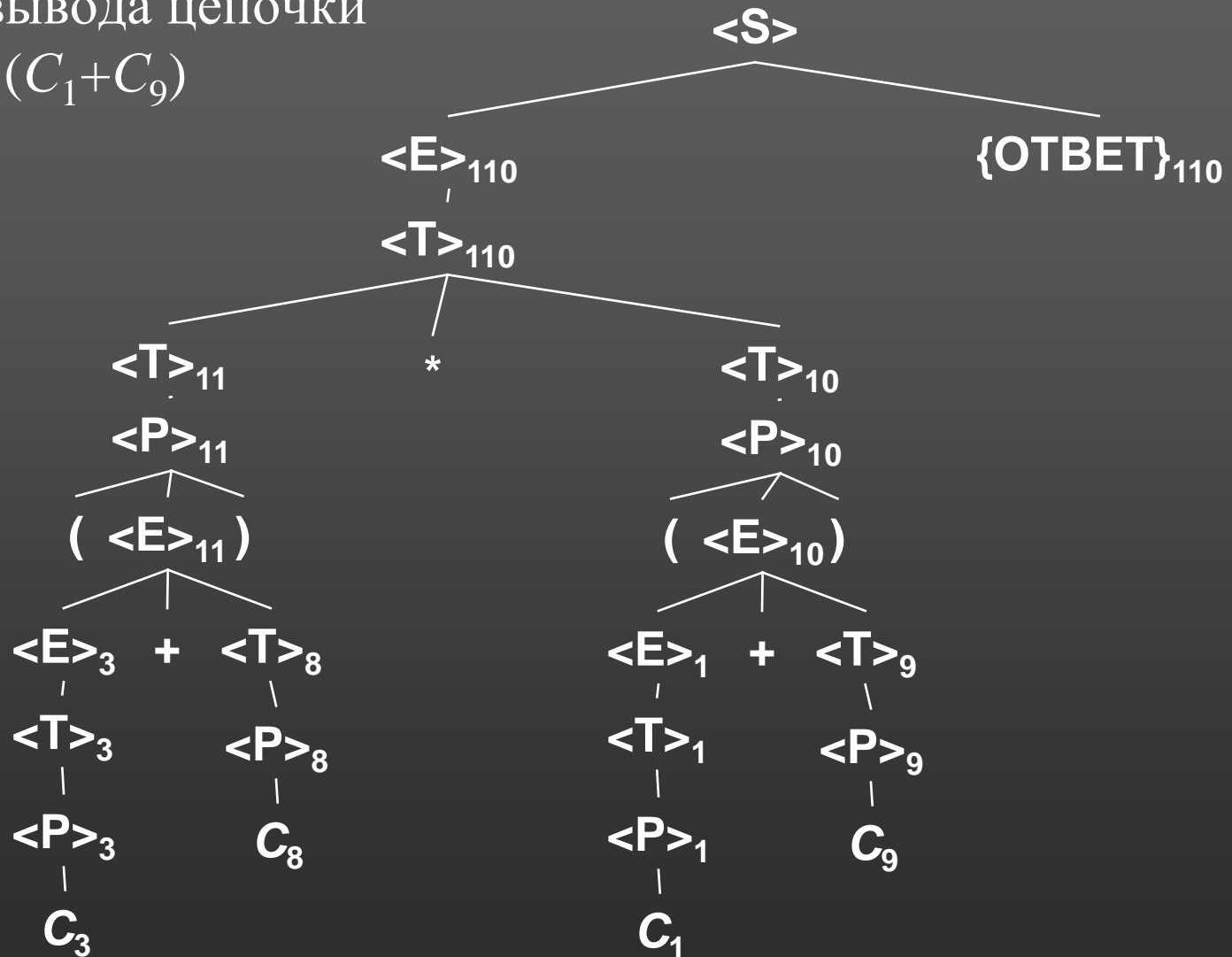
$p \leftarrow q$

7. $\langle P \rangle_p \rightarrow C_q$

$p \leftarrow q$

Пример синтезируемых атрибутов

- Дерево вывода цепочки $(C_3+C_8)*(C_1+C_9)$



Пример наследуемых атрибутов

- Как построить синтаксический блок, который по описанию переменных устанавливает соответствие между переменной и ее типом?
- Грамматика описания переменных:
 1. $\langle \text{описание} \rangle \rightarrow V \langle \text{список переменных} \rangle : \text{ТИП}$
 2. $\langle \text{список переменных} \rangle \rightarrow , V \langle \text{список переменных} \rangle$
 3. $\langle \text{список переменных} \rangle \rightarrow \varepsilon$

ТИП – Integer | Real | Boolean | ...

V – указатель на элемент таблицы имен

- Используем процедуру
УСТ_ТИП(указатель, тип)

Пример наследуемых атрибутов

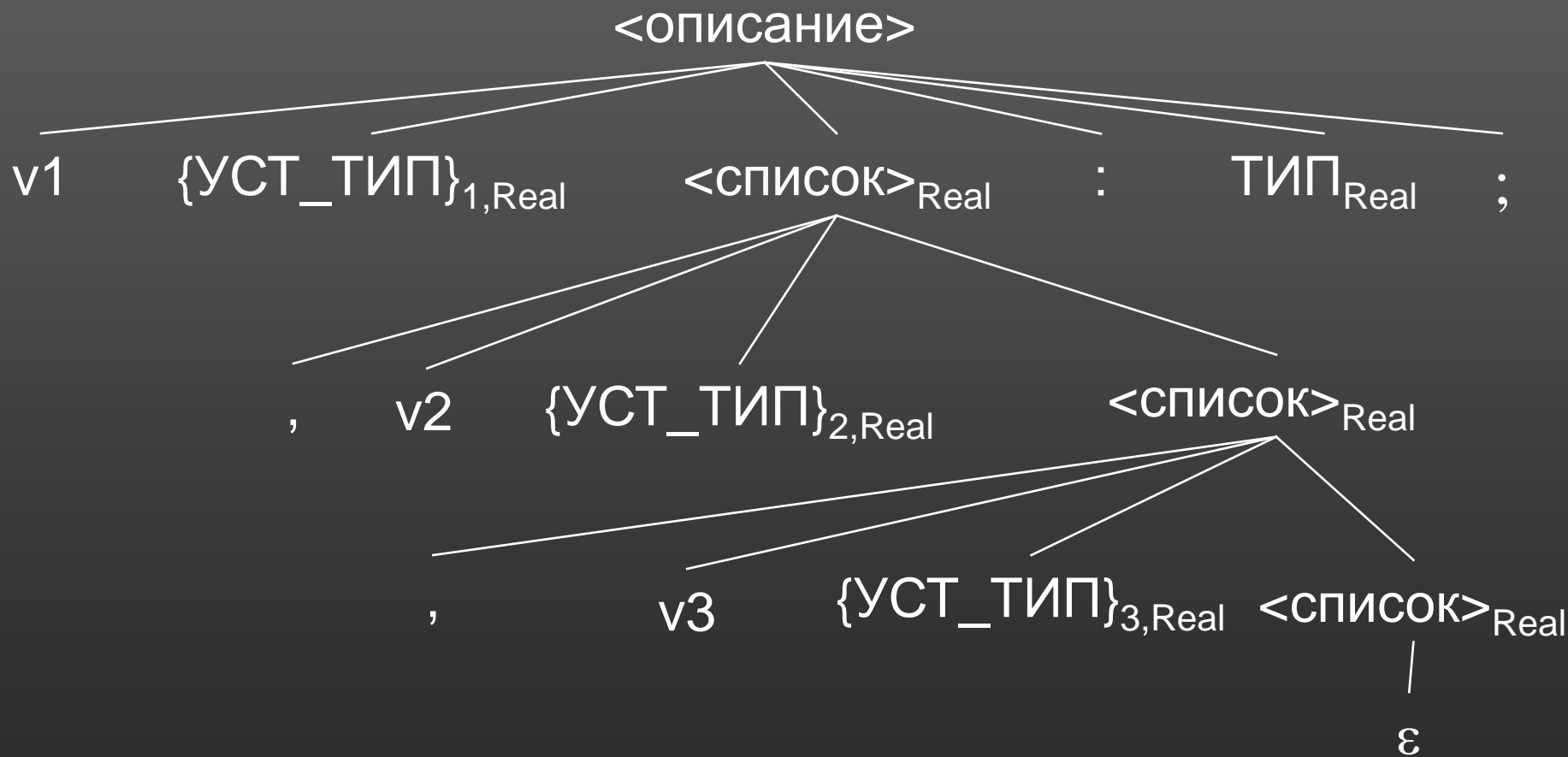
- Грамматика описания переменных:
 1. $\langle \text{описание} \rangle \rightarrow V \langle \text{список} \rangle : \text{ТИП}$
 2. $\langle \text{список} \rangle \rightarrow , V \langle \text{список} \rangle$
 3. $\langle \text{список} \rangle \rightarrow \varepsilon$
- Грамматика перевода описания переменных:
 1. $\langle \text{описание} \rangle \rightarrow V \{ \text{УСТ_ТИП} \} \langle \text{список} \rangle : \text{ТИП}$
 2. $\langle \text{список} \rangle \rightarrow , V \{ \text{УСТ_ТИП} \} \langle \text{список} \rangle$
 3. $\langle \text{список} \rangle \rightarrow \varepsilon$

Пример наследуемых атрибутов

- Грамматика перевода описания переменных:
 1. $\langle \text{описание} \rangle \rightarrow V \{ \text{УСТ_ТИП} \} \langle \text{список} \rangle : \text{ТИП}$
 2. $\langle \text{список} \rangle \rightarrow, V \{ \text{УСТ_ТИП} \} \langle \text{список} \rangle$
 3. $\langle \text{список} \rangle \rightarrow \varepsilon$
- Введем атрибуты **p** (указатель) и **t** (тип) для символа действия {УСТ_ТИП}. Атрибутная грамматика описания переменных:
 1. $\langle \text{описание} \rangle \rightarrow V_p \{ \text{УСТ_ТИП} \}_{p1,t1} \langle \text{список} \rangle_{t2} : \text{ТИП}_t$
 $t1 \leftarrow t, t2 \leftarrow t, p1 \leftarrow p$
 2. $\langle \text{список} \rangle_t \rightarrow, V_p \{ \text{УСТ_ТИП} \}_{p1,t1} \langle \text{список} \rangle_{t2}$
 $t1 \leftarrow t, t2 \leftarrow t, p1 \leftarrow p$
 3. $\langle \text{список} \rangle_t \rightarrow \varepsilon$

Пример наследуемых атрибутов

- Дерево вывода для описания переменных $v_1, v_2, v_3 : \text{Real}$;



Перевод арифметических выражений

- Задача: построить атрибутивную транслирующую грамматику, которая описывает обработку арифметических выражений синтаксическим блоком компилятора.
- Пример использования искомой грамматики:
 - Выражение: $(a+b)*(a+c)$
 - Входная цепочка лексем: $(I_A+I_B)*(I_A+I_C)$
 - Выходная цепочка атомов:
СЛОЖ(A,B,R1) СЛОЖ(A,C,R2) УМНОЖ(R1,R2,R3)
 - Таблица имен:
A – индекс идентификатора a , B – индекс идентификатора b ,
C – индекс идентификатора c , R1, R2, R3 – индексы промежуточных результатов

Перевод арифметических выражений

- Введем символы действия {СЛОЖ} и {УМНОЖ}, которые соответствуют атомам СЛОЖ и УМНОЖ, и построим грамматику перевода (аналогично грамматике постфиксной записи):

$$1. \langle E \rangle \rightarrow \langle E \rangle + \langle T \rangle \{ \text{СЛОЖ} \}$$

$$2. \langle E \rangle \rightarrow \langle T \rangle$$

$$3. \langle T \rangle \rightarrow \langle T \rangle * \langle P \rangle \{ \text{УМНОЖ} \}$$

$$4. \langle T \rangle \rightarrow \langle P \rangle$$

$$5. \langle P \rangle \rightarrow (\langle E \rangle)$$

$$6. \langle P \rangle \rightarrow I$$

Перевод арифметических выражений

- Введем атрибуты и правила их вычисления, чтобы полученная грамматика стала грамматикой перевода и позволяла вычислять значения выходных символов:
 - синтезируемые – один для каждого нетерминала, обозначает индекс элемента таблицы имен, который указывает на выражение, порождаемое данным нетерминалом
 - наследуемые – три для каждого символа действия, обозначают левый операнд, правый операнд и результат операции
- Введем системную подпрограмму НОВТ, которая выдает индекс некоторого неиспользованного элемента таблицы имен.

Перевод арифметических выражений

1. $\langle E \rangle \rightarrow \langle E \rangle + \langle T \rangle \{ \text{СЛОЖ} \}$
2. $\langle E \rangle \rightarrow \langle T \rangle$
3. $\langle T \rangle \rightarrow \langle T \rangle * \langle P \rangle \{ \text{УМНОЖ} \}$
4. $\langle T \rangle \rightarrow \langle P \rangle$
5. $\langle P \rangle \rightarrow (\langle E \rangle)$
6. $\langle P \rangle \rightarrow I$



1. $\langle E \rangle_x \rightarrow \langle E \rangle_q + \langle T \rangle_r \{ \text{СЛОЖ}_{y,z,p} \}$
 $(x,p) \leftarrow \text{НОВТ} \quad y \leftarrow q \quad z \leftarrow r$
2. $\langle E \rangle_x \rightarrow \langle T \rangle_p$
 $x \leftarrow p$
3. $\langle T \rangle_x \rightarrow \langle T \rangle_q * \langle P \rangle_r \{ \text{УМНОЖ}_{y,z,p} \}$
 $(x,p) \leftarrow \text{НОВТ} \quad y \leftarrow q \quad z \leftarrow r$
4. $\langle T \rangle_x \rightarrow \langle P \rangle_p$
 $x \leftarrow p$
5. $\langle P \rangle_x \rightarrow (\langle E \rangle_p)$
 $x \leftarrow p$
6. $\langle P \rangle_x \rightarrow I_p$
 $x \leftarrow p$

Перевод оператора

- Для перевода оператора присваивания вида **идентификатор := арифметическое выражение** нужно дополнить грамматику перевода арифметических выражений правилом
 $\langle \text{оператор} \rangle \rightarrow I_{var1} := \langle E \rangle_{expr1} \{ \text{ПРИСВОИТЬ}_{var2, expr2} \}$
 $var2 \leftarrow var1 \quad expr2 \leftarrow expr1$

Заключение

- Для задания **КС-грамматики** нужно определить конечное множество **терминалов**, конечное множество **нетерминалов**, конечное множество **правил (продукций)** и начальный нетерминал.
- **КС-язык** состоит из цепочек, которые можно получить из начального нетерминала КС-грамматики. **Класс КС-языков мощнее, чем класс регулярных множеств**: любое регулярное множество можно описать с помощью КС-грамматики; конечный распознаватель КС-языка можно построить только в том случае, если правила КС-грамматики имеют специальный вид.

Заключение

- **Синтаксически управляемая обработка** КС-языка означает обработку каждого отдельного правила соответствующей КС-грамматики.
- **Транслирующая грамматика (грамматика перевода)** – КС-грамматика, терминалы которой разбиты на два подмножества: **ВХОДНЫЕ СИМВОЛЫ** и **СИМВОЛЫ ДЕЙСТВИЯ**.
- Рассмотрена задача построения грамматики перевода арифметических выражений в **польскую запись**, в которой знак операции помещается *после* операндов.

Заключение

- **Атрибутные транслирующие грамматики** позволяют, помимо перевода цепочек, вычислять их значение в некотором смысле.
- Входные символы, символы действия и нетерминалы атрибутной транслирующей грамматики обладают конечным множеством **атрибутов** – свойств, которые используются для вычисления значения цепочки.
- Атрибуты делятся на **синтезируемые** и **наследуемые**.