

ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО
ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
УФИМСКИЙ ГОСУДАРСТВЕННЫЙ АВИАЦИОННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Кафедра ТК

курс лекций по дисциплине

Методы построения трансляторов

Тема: Грамматика языка

Преподаватель: к.т.н., доцент Карамзина А.Г.

Тема № 4

Грамматика языка

- Определение грамматики языка
- Запись правил грамматик в форме Бэкуса-Наура
- Классификация грамматик и языков

Определение грамматики языка

Грамматика – это описание способа построения предложений некоторого языка.

Правило (продукция) – это упорядоченная пара цепочек символов (α, β) .

В правилах очень *важен порядок* цепочек, поэтому их чаще записывают в виде

$$\alpha \rightarrow \beta$$

читается как « α порождает β » или « β по определению есть α ».

Грамматика языка программирования содержит правила двух типов:

- правила, определяющие синтаксические конструкции языка легко поддаются формальному описанию;
- правила, определяющие семантические ограничения языка обычно излагаются в неформальной форме.

Язык, заданный грамматикой \mathbf{G} , обозначается как $L(\mathbf{G})$.

Определение грамматики языка

Формально грамматика G определяется как четверка:

$$G(VT, VN, P, S),$$

где: VT – множество терминальных символов;

V

Множество терминальных символов VT

V^* ;

Множество нетерминальных символов VN содержит символы, которые определяют слова, понятия, конструкции языка

S

Множ
 G .

Рекурсия в правилах грамматики выражается в том, что один из нетерминальных символов определяется сам через себя.

Рекурсия может быть *непосредственной (явной)* – тогда символ определяется сам через себя в одном правиле,

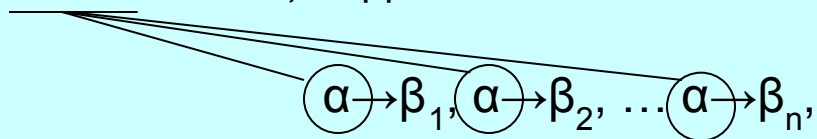
либо *косвенной (неявной)* – тогда символ определяется через цепочку правил.

Особенности
бесконечности
счет р

Запись правил грамматик в форме Бэкуса-Наура

Данная форма записи правил грамматик предполагает, что:

- если во множестве правил грамматики есть несколько правил, имеющих одинаковые левые части, вида:



то эти правила объединяют вместе и записываются в следующем виде:

$$\alpha \rightarrow \beta_1 | \beta_2 | \dots | \beta_n |$$

(одной строке в такой записи соответствует сразу n правил);

- нетерминальные символы берутся в угловые скобки: $\langle \rangle$.

Запись правил грамматики в форме Бэкуса-Наура

Пример: грамматика для целых десятичных чисел со знаком задана:

$G(\{0,1,2,3,4,5,6,7,8,9,-,+ \}, \{<число>, <чис>, <цифра>\}, P, <число>):$

$P: <число> \rightarrow <чис> \mid +<чис> \mid -<чис>$
 $<чис> \rightarrow <цифра> \mid <чис><цифра>$
 $<цифра> \rightarrow 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$

$G(\{0,1,2,3,4,5,6,7,8,9,-,+ \}, \{S, T, F\}, P, S):$

$P: S \rightarrow T \mid +T \mid -T$
 $T \rightarrow F \mid TF$
 $F \rightarrow 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$

держит двенадцать элементов:

вания нетерминальных символов
 е обязаны быть осмысленными,
 лано просто для удобства понимания
 правил грамматики человеком.
 же терминальных символов всегда
 строго соответствует алфавиту языка,
 определяемого грамматикой.

то есть имеются только
 рекурсия в 5 правиле,
 4 правило позволяет избежа

целевым символом грамматики является символ <число>.

Классификация грамматик и языков

Формальные грамматики классифицируются по структуре их правил.

Языки классифицируются в соответствии с типами грамматик.

От того, к какому типу относится тот или иной язык программирования, зависит сложность распознавателя для этого языка

чем сложнее язык, тем выше вычислительные затраты компилятора

Для некоторых типов языков в принципе невозможно построить компилятор, который анализировал бы исходные тексты на этих языках за приемлемое время на основе ограниченных вычислительных ресурсов (именно поэтому до сих пор невозможно создавать программы на естественных языках).

Классификация грамматик (по Хомскому)

Тип/Критерии		Грамматика
Тип 0		<i>с фразовой структурой</i>
правила		$\alpha \rightarrow \beta$, где $\alpha \in V^+$, $\beta \in V^*$
особенность		на структуру правил не накладывается никаких ограничений.
область применения		при построении компиляторов практического применения не имеют

Классификация грамматик (по Хомскому)

Тип/Критерии		Грамматика
Тип 1		контекстно-зависимые (КЗ)
правила	КЗ-грамматики	
	$\alpha_1 A \alpha_2 \rightarrow \alpha_1 \beta \alpha_2$, где $\alpha_1, \alpha_2 \in V^*$, $A \in VN$, $\beta \in V^+$	
	неукорачивающие КЗ-грамматики	
	$\alpha \rightarrow \beta$, где $\alpha, \beta \in V^+$, $ \beta \geq \alpha $	
особенность	КЗ-грамматики	
	при построении предложений языка, заданного грамматикой, один и тот же нетерминальный символ может быть заменен на ту или иную цепочку символов в зависимости от того контекста, в котором он встречается (<i>значение одного и того же символа может быть различным в зависимости от того, в каком контексте он встречается</i>). Цепочки α_1 и α_2 в правилах обозначают контекст (α_1 – левый контекст, α_2 – правый контекст) в общем случае любая из них или обе могут быть пустыми	
	неукорачивающие КЗ-грамматики	
	при построении предложений языка, заданного грамматикой, любая цепочка символов может быть заменена на цепочку символов не меньшей длины	
область применения	при построении компиляторов практического применения не имеют	

Классификация грамматик (по Хомскому)

Тип/Критерии		Грамматика
Тип 2		<i>контекстно-свободные (КС)</i>
	правила	<i>КС-грамматики</i>
		$A \rightarrow \beta$, где $A \in VN$, $\beta \in V^+$
		<i>укорачивающие КС-грамматики</i>
	особенность	$A \rightarrow \beta$, где $A \in VN$, $\beta \in V^*$
		<i>укорачивающие КС-грамматики</i> в правой части правил может присутствовать пустая цепочка
область применения	используются при описании синтаксических конструкций языков программирования	

Классификация грамматик (по Хомскому)

Тип/Критерии		Грамматика
Тип 3		<i>регулярные</i>
правила		<i>леволинейные</i>
		$A \rightarrow B\gamma$ или $A \rightarrow \gamma$, где $A, B \in VN, \gamma \in VT^*$.
		<i>праволинейные</i>
		$A \rightarrow \gamma B$ или $A \rightarrow \gamma$, где $A, B \in VN, \gamma \in VT^*$.
область применения	используются при описании простейших конструкций языков программирования: идентификаторов, констант, строк, комментариев и т.д.	

Классификация языков

Тип/Критерии		Язык
Тип 0		<i>с фразовой структурой</i>
задаются	грамматикой типа 0	
время распознавания	<u>невозможно</u> построить компилятор, который <u>гарантированно</u> выполнял бы разбор предложений языка за <u>ограниченное</u> время на основе <u>ограниченных</u> вычислительных ресурсов	
область применения	все естественные языки относятся к этому типу, поскольку структура и значение фразы естественного языка может зависеть не только от контекста данной фразы, но и от содержания того текста, где эта фраза встречается, поэтому существуют большие сложности в автоматизации перевода текстов, написанных на естественных языках и отсутствуют компиляторы, которые воспринимали бы программы на основе таких языков	

Классификация языков

Тип/Критерии		Язык
Тип 1		<i>контекстно-зависимые (КЗ)</i>
	задаются	грамматикой типа 1
	время распознавания	в общем случае время на распознавание предложений языка <u>экспоненциально</u> зависит от длины исходной цепочки символов
	область применения	применяются в анализе и переводе текстов на естественных языках; распознаватели, построенные на их основе, позволяют анализировать тексты с учетом контекстной зависимости в предложениях входного языка (<i>но не учитывают содержание текста, поэтому в общем случае для точного перевода с естественного языка все же требуется вмешательство человека</i>); также может выполняться автоматизированный перевод с одного естественного языка на другой, могут использоваться сервисные функции проверки орфографии и правописания в языковых процессорах.

Классификация языков

Тип/Критерии		Язык
Тип 2		контекстно-свободные (КС)
	задаются	грамматикой типа 2
	время распознавания	в общем случае время на распознавание предложений языка <u>полиномиально</u> зависит от длины исходной цепочки символов (<i>в зависимости от класса языка это либо кубическая, либо квадратичная зависимость, также существует много классов языков, для которых эта зависимость линейна</i>)
	область применения	лежат в основе синтаксических конструкций большинства современных языков программирования, на их основе функционируют некоторые довольно сложные командные процессоры, допускающие управляющие команды цикла и условия.

Классификация языков

Тип/Критерии		Язык
Тип 3		<i>регулярные</i>
	задаются	грамматикой типа 3
	время распознавания	время на распознавание предложений языка <u>линейно</u> зависит от длины исходной цепочки символов
	область применения	лежат в основе простейших конструкций языков программирования (<i>идентификаторов, констант и т.п.</i>), кроме того, на их основе строятся многие мнемокоды машинных команд (<i>языки ассемблеров</i>), а также командные процессоры, символьные управляющие команды и другие подобные структуры.