# Теория формальных языков и компиляторов
# Часть 1. Порождающие грамматики и языки

## Лекция 8. Классификация Хомского

При проектировании языковых процессоров трудно эмпирически подобрать грамматику, которая будет отвечать требованиям однозначности. Кроме того, не для всякой грамматики легко построить синтаксический анализатор. Частично решить эти проблемы можно, если знать некоторые особенности грамматики. Н. Хомский предложил способ классификации грамматик, способствующий решению проблем однозначности и безвозвратности разбора текста.

Классификация предполагает определение однозначности и безвозвратности разбора без построения синтаксических деревьев для одной и той же основы. С учётом классификации Хомского проект языкового процессора застрахован от неправильной работоспособности. Поэтому после построения грамматики её необходимо классифицировать. И в случае неудачной классификации необходимо изменить либо язык, либо грамматику так, чтобы классификация удовлетворяла требованиям однозначности и безвозвратности.

### 8.1 Классы грамматик

Н. Хомский выделил *четыре класса грамматик*: грамматики нулевого типа, контекстно-зависимые, контекстно-свободные и автоматные грамматики. **Вид грамматики определяется исходя из формы записи ее правил.** Рассмотрим классы грамматик.

**Определение 8.1. Грамматики нулевого типа** имеют правила следующего вида:

$$\alpha \to \beta,$$

где $\alpha \in V^*$, $V=V_T \cup V_N$, $\beta \in V^*$.

То есть в правой и левой частях правила находятся любые последовательности терминальных и нетерминальных символов алфавита. Этому требованию отвечают любые правила, которые только можно придумать. Значит, все возможные грамматики можно отнести к нулевому типу.

**Пример 8.1.** Грамматика G[A] имеет только одно правило: $Aab \rightarrow cBd$. Оно соответствует определению 8.1, значит G[A] – грамматика нулевого типа.

Грамматики нулевого типа не имеют практического применения. Этот класс грамматик является неоднозначным.

**Определение 8.2. Контекстно-зависимые грамматики** (КЗ-грамматики) имеют правила следующего вида:

$$\gamma_1 A \gamma_2 \rightarrow \gamma_1 \beta \gamma_2,$$

где $A \in V_N$, $\beta \in V^+$, $\gamma_1 \in V^*$ и $\gamma_2 \in V^*$.

Здесь в левой части правила есть нетерминальный символ A, который заменяется на непустую последовательность $\beta$, состоящую из терминалов и нетерминалов. Обратите внимание, что это более строгое требование к форме записи правил грамматики, чем в определении 8.1. Правило из примера 8.1 не соответствует этому определению, поэтому G[A] не является КЗ-грамматикой.

**Определение 8.3.** Также можно использовать другое определение КЗ-грамматик, когда форма записи правил имеет вид:

$$\alpha \rightarrow \beta,$$

где $\alpha, \beta \in V^+$, $|\alpha| \leq |\beta|$.

**Пример 8.2.** Грамматика G[ S ] имеет правила:

G[ S ] ={ 1. $S \rightarrow aSBC \mid abC$

2. $CB \rightarrow BC$

3. $bB \rightarrow bb$

4. $bC \rightarrow bc$

5. $cC \rightarrow cc\}$.

Данная грамматика является контекстно-зависимой, поскольку правила записаны в соответствии с определением 8.3. Кроме того, эту грамматику можно отнести и к нулевому типу, поскольку правила соответствуют определению 8.1. Поэтому при определении типа грамматики окончательно выбирается тот класс, для которого выполняются самые строгие ограничения на вид правил грамматики. В данном случае G[S] является КЗ-грамматикой.

Контекстно-зависимые грамматики в общем случае также неоднозначны. Они имеют ограниченное применение только в тех случаях, когда можно доказать однозначность конкретной грамматики.

**Определение 8.4.** Контекстно-свободные грамматики имеют правила следующего вида:

$$A \rightarrow \alpha,$$

где $A \in V_N$, $\alpha \in V^*$.

Здесь в левой части правил может быть только один нетерминальный символ, а в правой – любая последовательность из терминалов и нетерминалов. Это ограничение на вид правил еще более строгое, чем для контекстно-зависимых грамматик.

**Пример 8.3.** G[ R ]:

1. $R \rightarrow aa$

2. $R \rightarrow aAa$

3. $A \rightarrow b$

4. $A \rightarrow bA$

Контекстно-свободные грамматики или КС-грамматики имеют гораздо более широкое применение, в отличие от предыдущих типов грамматик. Однако доказано, что в общем случае нельзя показать однозначность и

безвозвратность КС-грамматик. Поэтому эти грамматики имеют также ограниченное применение. В индустрии проектирования процессоров широко используются подклассы КС-грамматик, для которых однозначность доказана. Например, S-грамматики.

**Определение 8.5.** Автоматные или регулярные грамматики имеют самые строгие ограничения на форму записи правил:

$$A \rightarrow aB \mid a \mid \varepsilon,$$

где $a \in V_T$, $A \in V_N$ и $B \in V_N$.

Здесь в левой части правил может быть только один нетерминальный символ, а в правой – либо один терминальный и один нетерминальный символ, либо один терминальный символ, либо пустая строка.

Для этого класса однозначность и безвозвратность доказана. Поэтому это наиболее часто используемый на практике тип грамматик.

**Пример 8.4.** G[I]:

1. $I \rightarrow aI$
2. $I \rightarrow bA$
3. $A \rightarrow bI$
4. $A \rightarrow a$

Определим, принадлежит ли цепочка «abbba» языку L(G[I])?

Для этого построим синтаксическое дерево (рис 8.1).

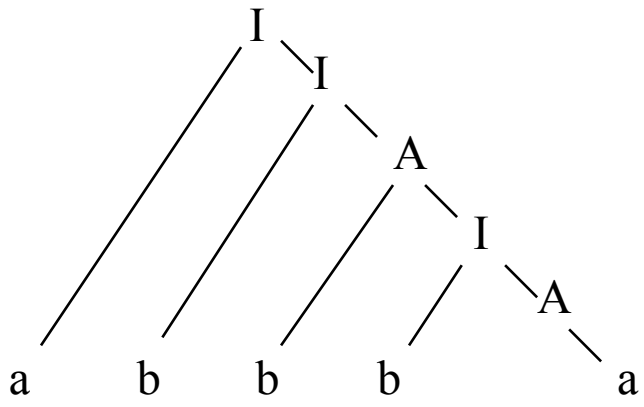



Рис. 8.1. Синтаксическое дерево

Следовательно, цепочка «abbba» принадлежит языку L(G[I]).

Из иллюстрации приведённого примера и особенностей правил вывода для автоматной грамматики следует, что структура синтаксического дерева этого класса грамматик имеет вид, показанный на рисунке 8.2.

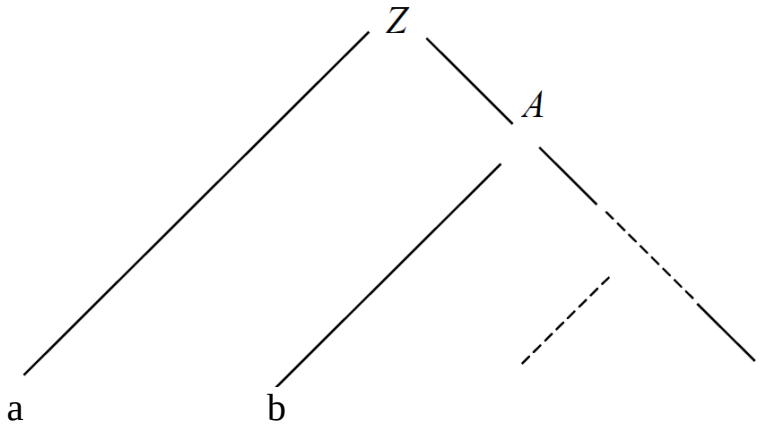



Рис. 8.2. Структура графа автоматной грамматики

Рассмотренный тип автоматных грамматик по определению является *праволинейным* типом автоматных грамматик. Существуют также и *леволинейные* автоматные грамматики. Продукции таких грамматик имеют вид^

$$A \to Ba \mid a \mid \varepsilon,\ a \in V_T,\ A \in V_N,\ B \in V_N.$$

Известен однозначный алгоритм перехода от леволинейной грамматики к праволинейной. Здесь мы не будем останавливаться на этом алгоритме, и в дальнейшем будем рассматривать только праволинейные грамматики. Тем более, что при проектировании грамматик праволинейные автоматные грамматики доминируют над леволинейными, и всегда можно избежать леволинейного проектирования грамматических конструкций

В заключении отметим, что приведённая классификация – включающая (рис. 8.3), т.е. все контекстно-свободные грамматики являются и контекстно-зависимыми, все контекстно-зависимые грамматики являются грамматиками общего вида и т.д. Кроме того, можно показать, что существуют языки, принадлежащие к типу i, но не к типу i+1. Например, как было показано, язык G[S] является контекстно-зависимым, но не контекстно-свободным.

Наконец, отметим, что определение контекстно-зависимой грамматики запрещает использование правил вида $A \rightarrow \varepsilon$.

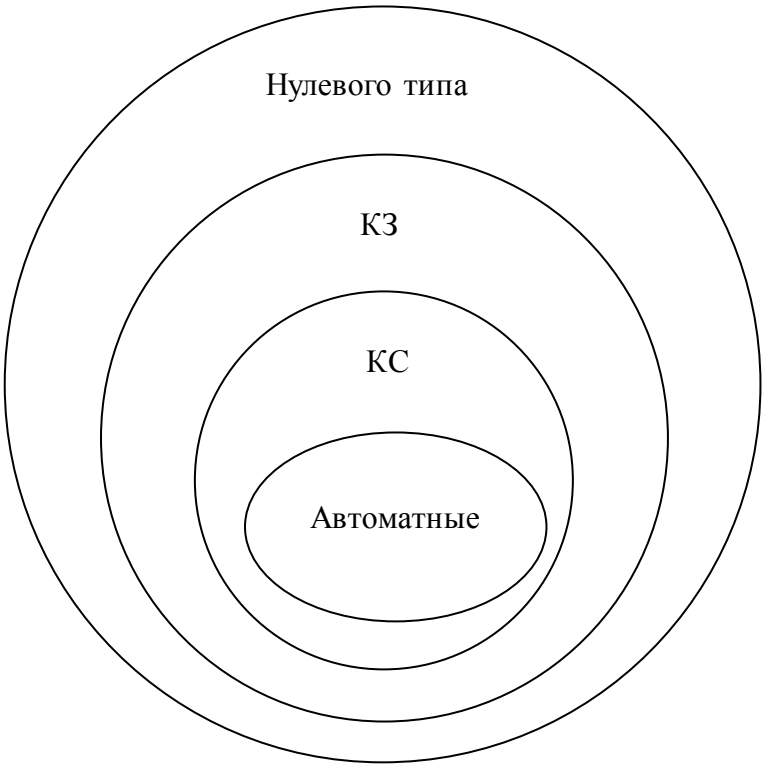



Рис. 8.3. Вложенность классов грамматик

## 8.2. Графы автоматных грамматик

Графы или диаграммы состояний также являются геометрической интерпретацией синтаксического анализа слева направо. В отличие от синтаксических синтаксических деревьев графы или диаграммы состояний используются для геометрической иллюстрации только автоматных грамматик.

**Определение 8.6.** Графом Г(G[Z]) или диаграммой состояний называется совокупность узлов и направленных дуг, соединяющих узлы. Узлы графа соответствуют нетерминальным символам. Дуги графа направлены из одного узла в другой (или в тот же узел) таким образом (рис. 8.4), что из узла N дуга направляется в M и маркируется терминалом t, если в грамматике имеется правило вида: $N \rightarrow t\ M$.

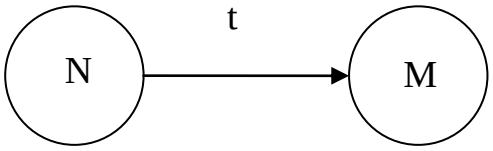



Рис. 8.4. Фрагмент графа

В этом случае имеет место отображение декартова произведения множества $V_N$ и множества $V_T$ на множество $V_N$, то есть

$$P: V_N \times V_T \to V_N .$$

Для правил вида $N \to t$ или $N \to \varepsilon$ следует ввести дополнительный нетерминал K, который определяет все заключительные узлы графа (рис. 8.5). Заключительные узлы изображаются с утолщенным или двойным контуром.

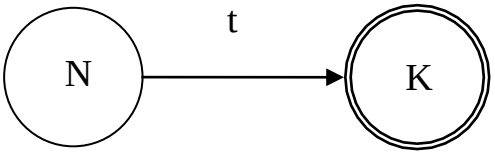



Рис. 8.5. Фрагмент графа с заключительным узлом

Проиллюстрируем процедуру построения графа для автоматной грамматики G[I] из примера 8.4:

1. $I \to aI$
2. $I \to bA$
3. $A \to bI$
4. $A \to a$

Введём в правило 4 нетерминал K, который определит заключительный узел на графе. Тогда G[I] будет иметь вид:

1. $I \to aI$
2. $I \to bA$
3. $A \to bI$
4. $A \to aK$
5. $K \to \varepsilon$.

Соответственно, терминальный и нетерминальный словари будут определены множествами:

$$V_T = \{a, b, \varepsilon\},\ V_N = \{I, A, K\}.$$

Граф Г(G[I]) или диаграмма состояний в соответствии с определением 8.6 будет иметь вид, представленный на рис. 8.6.

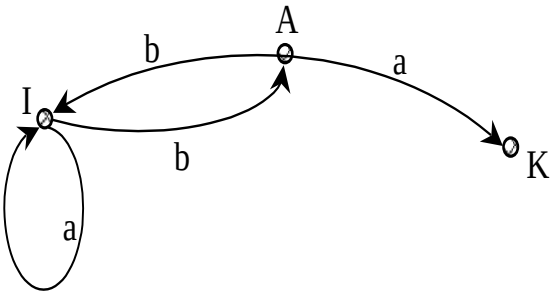


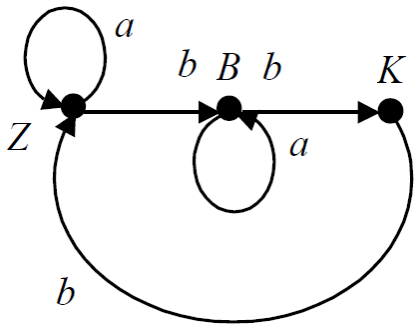

Рис.2.13. Граф Г(G[I]).

Диаграмма состояний, или граф, автоматически показывают принадлежность цепочки языку, порождаемому заданной грамматикой. Для установления факта принадлежности следует выполнить «путешествие» из начального нетерминала I в заключительное состояние K. Если такое «путешествие» завершается в узле K, то это означает принадлежность цепочки языку, порождаемому грамматикой G[I]. В противном случае цепочка не принадлежит языку.

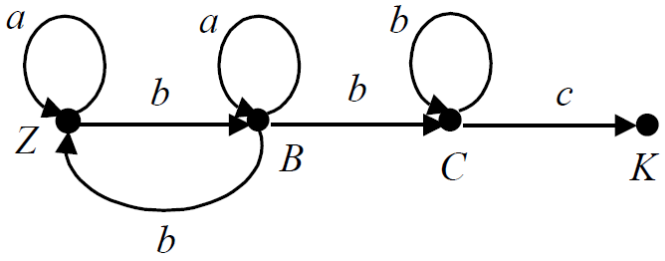
**Упражнения**

В №№ 1 – 4 подобрать язык L и грамматику G([Z]) по графу Г(G[Z]).

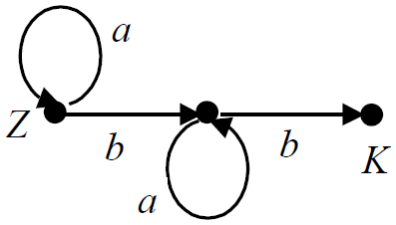
1.



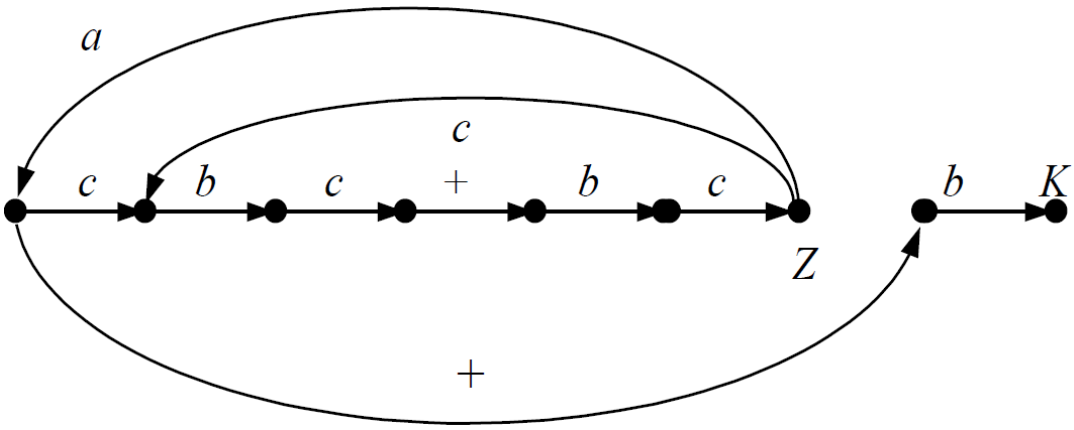
2.




3.




4.




**Список использованных источников**

1. Шорников Ю.В. Теория и практика языковых процессоров.