


# Классификация грамматик и языков

**Ноам Хомский**
**(Noam Chomsky)**

## 4 типа грамматик по Хомскому:

V+ — множество всех цепочек над алфавитом V без λ;
V* — множество всех цепочек над алфавитом V, включая λ.

**Тип 0: грамматики с фразовой структурой**

На структуру их правил не накладывается никаких ограничений: для грамматики вида G(VT,VN,P,S), $V = VN \cup VT$ правила имеют вид: $\alpha \rightarrow \beta$, где $\alpha \in V^+, \beta \in V^*$. Это самый общий тип грамматик. В него подпадают все без исключения формальные грамматики, но часть из них, к общей радости, может быть также отнесена и к другим классификационным типам. Дело в том, что грамматики, которые относятся только к типу 0 и не могут быть отнесены к другим типам, являются самыми сложными по структуре.

Практического применения грамматики, относящиеся только к типу 0, не имеют.

**Тип 1: контекстно-зависимые (КЗ) и неукорачивающие грамматики**

В этот тип входят два основных класса грамматик:

Контекстно-зависимые грамматики $G(VT,VN,P,S)$, $V = VN \cup VT$ имеют правила вида: $\alpha_1 A \alpha_2 \to \alpha_1 \beta \alpha_2$, где $\alpha_1, \alpha_2 \in V^*, A \in VN, \beta \in V^+$.

Неукорачивающие грамматики $G(VT,VN,P,S)$, $V = VN \cup VT$ имеют правила вида: $\alpha \to \beta$, где $\alpha, \beta \in V^+, |\beta| \geq |\alpha|$.

При построении предложений КЗ-грамматик один и тот же нетерминальный символ может быть заменен на ту или иную цепочку символов в зависимости от того контекста, в котором он встречается.

Цепочки α1 и α2 в правилах грамматики обозначают контекст (α1— левый контекст, а α2 — правый контекст), в общем случае любая из них (или даже обе) может быть пустой. Говоря иными словами, значение одного и того же символа может быть различным в зависимости от того, в каком контексте он встречается.

При построении компиляторов такие грамматики не применяются

Неукорачивающие грамматики имеют такую структуру правил, что при построении предложений языка, заданного грамматикой, любая цепочка символов может быть заменена на цепочку символов не меньшей длины.

**Тип 2: контекстно-свободные (КС) грамматики**

Контекстно-свободные (КС) грамматики $G(VT,VN,P,S)$, $V = VN \cup VT$ имеют правила вида: $A \to \beta$, где $A \in VN, \beta \in V^+$ . Такие грамматики также иногда называют неукорачивающими контекстно-свободными (НКС) грамматиками (видно, что в правой части правила у них должен всегда стоять как минимум один символ).

КС-грамматики широко используются при описании синтаксических конструкций языков программирования. Синтаксис большинства известных языков программирования основан именно на КС-грамматиках

**Тип 3: регулярные грамматики**

*К* типу регулярных относятся два эквивалентных класса грамматик: леволинейные и праволинейные.

Леволинейные грамматики G(VT,VN,P,S), $V = VN \cup VT$ могут иметь правила двух видов: $A \to B\gamma$ или $A \to \gamma$, где $A, B \in VN, \gamma \in VT^*$.

В свою очередь, праволинейные грамматики G(VT,VN,P,S), $V = VN \cup VT$ могут иметь правила тоже двух видов: $A \to \gamma B$ или $A \to \gamma$, где $A, B \in VN, \gamma \in VT^*$.

Регулярные грамматики используются при описании простейших конструкций языков программирования: идентификаторов, констант, строк, комментариев и т. д.

Для классификации грамматик всегда выбирают максимально возможный тип, к которому она может быть отнесена. Сложность грамматики обратно пропорциональна номеру типа, к которому относится грамматика. Грамматики, которые относятся только к типу 0, являются самыми сложными, а грамматики, которые можно отнести к типу 3 — самыми простыми.

# Классификация языков

## Тип 0: языки с фразовой структурой

Это самые сложные языки, которые могут быть заданы только грамматикой, относящейся к типу 0. Если язык относится к типу 0, то для него невозможно построить компилятор, который гарантированно выполнял бы разбор предложений языка за ограниченное время на основе ограниченных вычислительных ресурсов.

К сожалению, все естественные языки относятся к фразовым. Структура и значение фразы естественного языка может зависеть не только от контекста данной фразы, но и от содержания того текста, где эта фраза встречается. Одно и то же слово в естественном языке может не только иметь разный смысл, в зависимости от контекста, но и играть различные роли в предложении. Именно поэтому столь велики сложности в автоматизации перевода текстов, написанных на естественных языках

## Тип 1: контекстно-зависимые (КЗ) языки

Тип 1 — второй по сложности тип языков. В общем случае время на распознавание предложений языка, относящегося к типу 1, экспоненциально зависит от длины исходной цепочки символов.

**Языки и грамматики, относящиеся к типу 1, 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 (но они не учитывают содержание текста, поэтому для точного перевода с естественного языка требуется вмешательство человека). На основе таких грамматик может выполняться автоматизированный перевод с одного естественного языка на другой, ими могут пользоваться сервисные функции проверки орфографии и правописания в языковых процессорах.**

В компиляторах КЗ-языки не используются

**Тип 2: контекстно-свободные (КС) языки**

КС-языки лежат в основе синтаксических конструкций большинства современных языков программирования,

**Тип 3: регулярные языки**

Регулярные языки — самый простой тип языков. Поэтому они являются самым широко используемым типом языков в области вычислительных систем. Время на распознавание предложений регулярного языка линейно зависит от длины входной цепочки символов.

Регулярные языки лежат в основе простейших конструкций языков программирования (идентификаторов, констант и т. п.), кроме того, на их основе строятся языки ассемблеров, а также командные процессоры, символьные управляющие команды и другие подобные структуры.

## Чем все это безобразие распознавать

Для *языков с фразовой структурой* (тип 0) необходим распознаватель, имеющий неограниченную внешнюю память. Поэтому для языков данного типа нельзя гарантировать, что за ограниченное время на ограниченных вычислительных ресурсах распознаватель завершит работу и примет решение о том, принадлежит или не принадлежит входная цепочка заданному языку. Практического применения языки с фразовой структурой не имеют.

Для *контекстно-зависимых языков* (тип 1) распознавателями являются двусторонние недетерминированные автоматы с ограниченной памятью. Количество шагов, необходимых автомату для распознавания входной цепочки, экспоненциально зависит от длины этой цепочки.

Экспоненциальная зависимость времени разбора от длины цепочки существенно ограничивает применение распознавателей для контекстно-зависимых языков. Такие распознаватели применяются для автоматизированного перевода и анализа текстов на естественных языках, когда временные ограничения на разбор текста несущественны.

Для *контекстно-свободных языков* (тип 2) распознавателями являются односторонние недетерминированные автоматы с магазинной (стековой) внешней памятью — МП-автоматы. При простейшей реализации алгоритма работы такого автомата он имеет экспоненциальную сложность, однако путем некоторых усовершенствований алгоритма можно добиться полиномиальной (кубической) зависимости времени, необходимого на разбор входной цепочки, от длины этой цепочки. Следовательно, можно говорить о полиномиальной сложности распознавателя для КС-языков.

## Пример: грамматика целых десятичных чисел

G1{0,1,2,3,4,5,6,7,8,9,-,+},{S, T, F},P1,S):
P1:
S → T | +T | -T
T → F | TF
F → 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

По структуре своих правил данная грамматика G1 относится к контекстно-свободным грамматикам (тип 2). Ее можно отнести и к типу 0, и к типу 1, но максимально возможным является именно тип 2, поскольку к типу 3 эту грамматику отнести никак нельзя:
строка T → F | TF содержит правило T → TF, которое недопустимо для типа 3, и хотя все остальные правила этому типу соответствуют, одного несоответствия достаточно.

## Та же грамматика, но по-другому:

G1' ({0,1,2,3,4,5,6,7,8,9,-,+},{S, T},P1',S):
P1':
S → T | +T | -T
T → 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
0T | 1T | 2T | 3T | 4T | 5T | 6T | 7T | 8T | 9T

По структуре своих правил данная грамматика G1 является праволинейной и относится к типу 3.

Та же грамматика, но леволинейная:

G1" ({0,1,2,3,4,5,6,7,8,9,-,+},{S, T},P1",S):
P1':
T → + | - | λ
S → T0 | T1 | T2 | T3 | T4 | T5 | T6 | T7 | T8 | T9 | S0 | S1 | S2 | S3 | S4 | S5 | S6 | S7 | S8 | S9