**Ордена Трудового Красного Знамени**

**Федеральное государственное бюджетное образовательное учреждение**

**Высшего профессионального образования**

**Московский технический университет связи и информатики**

-\*\*-

**Реферат**

**по дисциплине**

**«Теоретические аспекты программирования»**

**по теме:**

**«Формальные языки»**

Выполнил:

магистрант гр. 3МПП1901

Иванюк П. Г.

Проверил:

Профессор, д.т.н., Яшина М. В.

**Москва 2020**

**Оглавление**

[**Введение** 3](#_Toc58175279)

[**Формальные языки** 4](#_Toc58175280)

[**Виды грамматик** 5](#_Toc58175281)

[**Формальная грамматика** 5](#_Toc58175282)

[**Классификация грамматик и языков.** 7](#_Toc58175283)

[**Заключение** 8](#_Toc58175284)

# **Введение**

Лингвистика – наука о языке. Математическая лингвистика – наука, занимающаяся формальными методами построения и изучения языков. Теория формальных грамматик – это раздел математической лингвистики, включающий способы описания формальных грамматик, построение алгоритмов анализа, а также алгоритмов перевода алгоритмических языков на язык машины.

Рассмотрим некоторые понятия из естественных языков.

Язык — это множество предложений, построенных по определенным правилам.

Грамматика — свод правил, определяющих принадлежность фразы языку.

Любой язык должен удовлетворять свойствам разрешимости и однозначности. Язык разрешим, если за конечное время можно определить, что фраза или предложение принадлежит языку.

Язык однозначен, если любая фраза понимается единственным образом.

Синтаксис — свод правил, определяющих правильность построения предложений языка.

Семантика — свод правил, определяющих семантическую или смысловую правильность предложений языка.

**Естественные языки**

Предложение может быть синтаксически верным и семантически неверным. Синтаксис обычно упрощается тем, что не все фразы языка обязаны иметь смысл.

<Предложение>

<группа подлежащего> <группа сказуемого>

<определение> <подлежащее> <образ действия> <сказуемое>

<прилагательное> <существительное> <местоимение> <наречие> глагол>

шоколадный пень быстро бежит

Фраза на русском языке, так как её можно разобрать по членам предложения, но смысл её неясен.

Теория формальных языков наиболее развитая ветвь математической лингвистики, которая является методикой объяснения языка машине.

Чтобы объяснить язык машине, необходимо четко представлять, как он устроен и как его понимать. Чтобы создать транслятор алгоритмического языка, необходимо иметь алгоритм перевода текста в те действия, которые следует выполнить, что в свою очередь, требует формализации языка. Алгоритмические языки создаются в расчете на формализацию и должны быть однозначными.

# **Формальные языки**

Формальный язык — это математическая абстракция, возникшая как обобщение обычных лингвистических понятий естественных языков. Теория формальных языков изучает синтаксис языков и является фундаментом синтаксически управляемых процессов перевода, к которому можно отнести трансляцию, ассемблирование и компиляцию. Основы этой теории были заложены американским математиком и лингвистом Ноамом Хомским (Chomsky) в конце 50-х – начале 60-х годов. Он предложил, каким образом описывать структуру языков программирования. Остановимся на основных элементах этой теории.

Тексты на любом языке (естественном или формальном) представляют собой цепочки символов некоторого алфавита.

Цепочкой называется произвольная, упорядоченная, конечная последовательность символов, записанных один за другим.

Операции над цепочками символов:

Конкатенация (объединение, сложение двух цепочек) – это дописывание второй цепочки в конец первой.

Замена (подстановка) – замена подцепочки символов на любую произвольную цепочку символов.

Обращение – запись символов цепочки в обратном порядке.

Итерация – повторение цепочки n раз.

Пустая цепочка – это цепочка, не содержащая ни одного символа.

Основой любого языка является алфавит, определяющий набор допустимых символов языка.

Алфавит – это счётное множество допустимых символов языка и обозначается это множество буквой V.

Языком L над алфавитом V или (L(V)) называется некоторое счётное подмножество цепочек конечной длины из множества всех цепочек над алфавитом V.

Кроме алфавита язык предусматривает также правила построения допустимых цепочек. Символы могут объединяться в слова или лексемы, что значит элементарные конструкции языка и на их основе строятся более сложные конструкции – предложения. И те и другие являются цепочками символов, предусматривающие некоторые правила построения.

В общем случае язык можно определить тремя способами:

1. перечислением всех допустимых цепочек;

Большинство языков содержат бесконечное число допустимых цепочек и перечислить их просто невозможно.

1. указанием способа порождения цепочек или заданием грамматики;

Предусматривает некоторое описание правил, с помощью которых строятся цепочки.

1. определением метода распознавания цепочек.

Предусматривает построение некоторого логического устройства (распознавателя) – автомата, который на входе получает цепочку символов, а на выходе выдаёт ответ, принадлежит или нет эта цепочка заданному языку.

# **Виды грамматик**

Все грамматики подразделяются на три вида. Порождающие грамматики – этот грамматика, которая используется для генерации цепочек языков по требованию. Распознающие грамматики – это грамматики, которые представляют собой системы модели, которым на вход подаётся цепочка некоторого языка, а на выходе она выдает или «Да», если цепочка принадлежит языку или «Нет» если цепочка не принадлежит языку. Перечисляющие грамматики – это грамматики, которые выдают одну за другой все цепочки языка.

# **Формальная грамматика**

Формальная грамматика – это описание способа построения предложений некоторого языка. Описывается различными способами. Например, использованием формального описания грамматики, построенное на основе системы правил.

Правило – это упорядоченная пара цепочек, например (α, β). В правилах важен порядок цепочек, поэтому их чаще записывают в виде α → β, что значит «α порождает β».

Грамматика языка программирования содержит правила двух типов: первые определяют синтаксические конструкции языка и довольно легко поддаются формальному описанию; вторые определяют семантические ограничения языка и обычно излагаются в неформальной форме. Поэтому любое описание языка программирования обычно состоит из двух частей: вначале формально излагаются правила построения синтаксических конструкций, а потом на естественном языке даётся описание семантических правил.

Формальная грамматика определяется как G = <V, W, P, S>, где:

V – терминальный алфавит.

W – не терминальный алфавит.

P – множество правил, которое имеет вид множества подстановок вида α→β.

S – аксиома, начальный символ грамматики.

Терминальный алфавит – это конкретные слова из словаря.

Не терминальный алфавит – это имена синтаксических классов.

Символы терминального алфавита обозначаются малыми латинскими буквами: a, b, c.

Символы не терминального алфавита обозначаются большими латинскими буквами: A, B, C.

Цепочки, то есть последовательности символов, с точки зрения грамматики неделимы.

Словам соответствуют символы терминального алфавита, а предложение это цепочка. Язык – это множество цепочек.

Язык, заданный грамматикой G, обозначается как L(G). Две грамматики, G и G', называются эквивалентными, если они определяют один и тот же язык: L (G) = L (G').

Алфавиты терминальных символов и нетерминальных символов грамматики не пересекаются. Каждый символ в грамматике может быть либо терминальным, либо нетерминальным, но не может быть терминальным и нетерминальным одновременно. Целевой символ грамматики – это всегда нетерминальный символ.

Множество символов терминального алфавита (V) – это символы, которые входят в алфавит языка, порождаемого грамматикой. Как правило, символы из этого множества встречаются только в цепочках правых частей правил.

Множество символов нетерминального алфавита (W) – это символы, которые определяют слова, понятия, конструкции языка. Каждый символ этого множества может встречаться в цепочках левой и правой частей правил грамматики.

Во множестве правил грамматики может быть несколько правил, имеющих одинаковые левые части вида: α → β1, α → β2, …, α → βn. Эти правила можно объединить вместе и записать в виде: α → β1 | β2 | … | βn.

Такая форма записи правил грамматики называется формой Бэкуса-Наура. Форма Бэкуса-Наура предусматривает, что нетерминальные символы берутся в угловые скобки.

Грамматика, которая определяет язык целых десятичных чисел со знаком:

G = ({0, 1, 2, 3, 4, 5, 6, 7, 8, 9, –, +}, {<число>, <зч>, <цифра>}, Р, <число>)

Р: <число> → <зч> | +<зч> | –<зч>

<зч> → <цифра> | <зч> | <цифра>

<цифра> → 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

Составляющие элементы грамматики G:

- множество терминальных символов V содержит двенадцать элементов: десять десятичных цифр и два знака;

- множество нетерминальных символов W содержит три элемента: символы <число>, <зч> и <цифра>;

- множество правил содержит 16 правил, которые записаны в три строки, то есть имеется только три различные левые части правил;

- Целевым символом грамматики является символ <число>.

Если нетерминальные символы обозначить большими латинскими буквами, то получится следующая грамматика для языка целых десятичных чисел со знаком:

G' = ({0, 1, 2, 3, 4, 5, 6, 7, 8, 9, –, +}, {S, T, F}, Р, S)

Р: S → Т | +Т | –Т

T → F | TF

F → 0 | l | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

Теперь W = {S, T, F}. Язык, заданный грамматикой, не изменился – грамматики G и

G' эквивалентны.

Особенность рассмотренных выше формальных грамматик в том, что они позволяют определить бесконечное множество цепочек языка с помощью конечного набора, правил. Возможность пользоваться конечным набором правил достигается в такой форме записи за счёт рекурсивных правил. Рекурсия в правилах грамматики выражается в том, что один из нетерминальных символов определяется сам через себя. Рекурсия может быть непосредственной (явной) – когда символ определяется сам через себя в одном правиле, либо косвенной (неявной) когда символ определяется через цепочку правил.

В рассмотренной выше грамматике G непосредственная рекурсия присутствует в правиле <зч> → <зч> <цифра>, а в эквивалентной ей грамматике G' – в правиле Т → TF. Чтобы рекурсия не была бесконечной, для участвующего в ней нетерминального символа грамматики должны существовать также и другие правила, которые определяют его, минуя самого себя, и позволяют избежать бесконечного рекурсивного определения. Такими правилами являются <зч> → <цифра> – в грамматике G и Т → F – в грамматике G'.

# **Классификация грамматик и языков.**

Формальные грамматики классифицируются по структуре их правил. Если все без исключения правила грамматики удовлетворяют некоторой заданной структуре, то такая грамматика относится к определённому типу. Достаточно иметь в грамматике одно правило, не удовлетворяющее требованиям структуры правил, и она уже не попадает в заданный тип. Согласно классификации Хомского, выделяются четыре типа грамматик.

Типа 0: грамматики с произвольной структурой.

На структуру их правил не накладывается никаких ограничений:

для грамматики вида G = <V, W, P, S>, правила имеют вид α → β,

Типа 1: контекстно-зависимые грамматики или не укорачивающие грамматики.

для грамматики вида G = <V, W, P, S>, имеют правила вида αAβ → αγβ.

В этом правиле α и β не меняются, но их наличие необходимо, чтобы правило было применено. Вместо одного нетерминального символа подставляется некое, произвольное слово γ, но при этом должно быть присутствие α и β которые образуют контекст. Поэтому такие грамматики называются контекстными или не укорачивающимися, потому что в результате применения одного правила общая цепочка станет только длиннее.

Типа 2: бесконтекстные грамматики.

G = <V, W, P, S>, имеют правила вида А → α, где А один не терминальный символ, α произвольная цепочка.

Такие грамматики называются беcконтекстными, поскольку контекст уже не нужен. Эти грамматики называются контекстно-свободными (context-free). Данный тип грамматик лежит в основе описания теории языков программирования и трансляторов, потому что они обладают хорошими свойствами, достаточными для их описания.

Типа 3: регулярные грамматики.

Этот тип называется автоматным или регулярным.

G = <V, W, P, S>, имеют правила вида А→αВ или А→α

А – один не терминальный символ, αВ – один терминальный и один не терминальный символ либо один только терминальный символ.

Языки, которые порождаются типом 3 это те множества, которые распознаются конечными автоматами.

Для любого языка типа 3 если автомат, который его распознает и любое множество, которое распознаёт любой автомат является языком типа 3.

Классификация Хомского называется иерархия Хомского так как каждый следующий тип является частным случаем предыдущего.

Грамматики типа 0 эквивалентны машине Тьюринга. Любое перечислимое множество является языком типа 0.

Языки классифицируются в соответствии с типами грамматик, с помощью которых они заданы. Причём, поскольку один и тот же язык в общем случае может быть задан сколь угодно большим количеством грамматик, которые могут относиться к различным классификационным типам, для классификации самого языка среди всех его грамматик выбирается грамматика с максимально возможным классификационным типом.

# **Заключение**

Для описания семантики естественных языков используются конструкции построения предложений. Формальные языки предназначены для построения синтаксиса языков программирования, а также для построения их трансляторов.