

ПОНЯТИЕ ГРАММАТИКИ И ФОРМАЛЬНОЕ ОПРЕДЕЛЕНИЕ. ФОРМА БЭКУСА-НАУРА И ДРУГИЕ СПОСОБЫ ЗАДАНИЯ ГРАММАТИК

Фаст Владислав Викторович

*Студент ФГБОУ ВО «Армавирский государственный
педагогический университет», г. Армавир*

*Научный руководитель, к.т.н, доцент кафедры
информатики и ИТО Андрусенко Е.Ю.*

THE CONCEPT OF GRAMMAR AND FORMAL DEFINITION. THE BACKUS-NAUR FORM AND OTHER METHODS OF SPECIFYING GRAMMARS

Fast Vladislav Viktorovich

Student of Armavir state pedagogical University, Armavir

АННОТАЦИЯ

В данной научной работе рассмотрены грамматики и их формальное определение. Подробно разобраны некоторые формы задания грамматик, такие как: форма Бэкуса-Наура, форма запись правил грамматик с применением специальных символов и форма записи правил контекстно-свободных грамматик в графическом виде.

ABSTRACT

This research paper considers grammars and their formal definition. Some forms of grammar assignment are analyzed in detail, such as the Backus-Nour form, the form for writing grammar rules using special characters, and the form for writing rules for context-free grammars in graphical form.

Ключевые слова: Языки программирования, грамматика, формальные грамматики, форма Бэкуса-Наура, запись правил грамматик с применением специальных символов, запись правил контекстно-свободных грамматик в графическом виде.

Keyword: Programming languages, grammar, formal grammars, Backus-Naur form, writing grammar rules using special characters, writing context-free grammar rules in graphical form.

Языки программирования - это так называемый способ общения человека с Электронными Вычислительными машинами, ну или просто ЭВМ. Как и любой естественный язык, язык программирования имеет свой алфавит и свою грамматику. Алфавит - некоторое конечное и не пустое множество элементов (символов). Но для обозначений символов этого алфавита используются буквы латинского алфавита. Грамматика в целом представляет собой некое описание способов построения цепочек языка. А эти цепочки являются конечной последовательностью символов алфавита. Так как языки программирования относятся к формальным языкам необходимо дать определение самим формальным языкам. Итак, формальными языками называются произвольные множества тех самых цепочек, о которых мы говорили выше. Эти языки бываю конечными и бесконечными. В первом случае мы можем просто последовательно выписать все цепочки языка, но, когда язык бесконечен, необходимо использовать формальные грамматики.

Именно о грамматиках и формальных грамматиках мы хотим рассказать в своей научной работе.

Грамматикой языка называется способ задания языка, т.е. грамматикой мы можем назвать совершенно любой способ задания этого языка. К примеру, возьмём грамматику $K = \{a^n b^n\}$, где n - это некоторое натуральное число, задающее язык K , который состоит из следующих цепочек: ab , $aabb$, $aaabbb$ и т.п. Язык K - это скажем так, некоторое бесконечное множество таких цепочек, но его описание (грамматика) будет состоять всего лишь из 10 символов, из этого следует вывод, что она конечна.

На основе всего вышеописанного, можно смело сказать, что грамматика языка, описывает именно законы использования символов алфавита для построения входящих в него цепочек. Эти законы называются синтаксическими закономерностями.

Формой Бэкуса-Наура (БНФ), называется формальная система описания синтаксиса, где синтаксические категории одного типа можно определить через категории другого типа.

Говоря более простым языком, Форма Бэкуса-Наура является методом обозначения для контекстно-свободных грамматик, также БНФ применяют для того, чтобы описать синтаксис языка, который используется в вычислениях, например, в форматах документов, в компьютерных языках программирования, в связях протоколов, и наборе команд.

Но основным назначением контекстно-свободных грамматик является их применение для более точного описания языков, таких как:

- Учебники по теории языков программирования
- Руководства пользователя
- Официальные спецификации языка.

В большинстве правил контекстно-свободных грамматик возможно использование нескольких правил, у которых левые части одинаковые и имеют следующий вид: $\alpha \rightarrow \beta_1, \alpha \rightarrow \beta_2, \dots, \alpha \rightarrow \beta_n$. Получается, что при объединении этих правил запись будет выглядеть так: $\alpha \rightarrow \beta_1 | \beta_2 | \dots | \beta_n$.

Данный вид записи правил контекстно-свободных грамматик принято называть формой Бэкуса-Наура.

Эту форму записи правил контекстно-свободных грамматик разработал американский учёный в области информатики, создатель первого высокоуровневого языка программирования Фортран Джон Бэкус в 1959

А после доработана датский учёный в области информатики, создатель первого языка струнного программирования Алгол 60 Питером Науром.

По прошествии времени форма Бэкуса-Наура была модифицирована, в неё добавили несколько новых правил описания синтаксиса языка, такая форма была названа расширенной формой Бэкуса-Наура (РБНФ).

Запись правил грамматик с применением специальных символов

При записи правил контекстно-свободных грамматик с применением специальных символов или метасимволов использование подразумевает специальных символов в строке правил грамматики. Каждый из этих символов может иметь свой «особенный» смысл и объясняться по-своему.

К числу специальных символов (метасимволов) можно отнести:

- Круглые скобки - (), значат то, что из ряда всех перечисленных цепочек символов, находящихся внутри них, в определённом месте

правил контекстно-свободной грамматики, возможно использование только лишь одной цепочки;

- Квадратные скобки - $[]$ – значат то, что определённая цепочка, находящаяся в них, способна как не встречаться, так и встречаться в определённом месте правила контекстно-свободной грамматики;
- Фигурные скобки - $\{\}$ - значат то, что цепочка находящаяся внутри их может встречаться много раз, один раз или не встречаться вовсе;
- Запятые - «,» - Позволяют разделять символы цепочки, находящиеся в круглых скобках;
- Кавычки - «» или “” - Применяются в таких моментах, когда один из специальных символов необходимо внести в цепочку.

Запись правил контекстно-свободных грамматик в графическом виде

Главная суть использования записи правил контекстно-свободных грамматик в графическом виде заключается в построении диаграмм специальным образом. Данный вид записи был создан для того, чтобы описывать контекстно-свободные грамматики такого языка программирования, как Pascal, но немного позже этот вид записи стал распространён и в литературе. К сожалению, такая форма записи подходит не для всех типов контекстно-свободных грамматик, а только для описания более знаменитых языков программирования.

Эта форма записи правил, делает следующее: «Каждому нетерминальному символу соответствует своя диаграмма, которая построена в виде направленного графа»

Такие графы бывают следующих видов:

- Точка входа

- Нетерминальный символ
- Цепочка терминальных символов
- Узловая цепочка
- Точка выхода.

Список литературы:

1. https://kryukova.engineer-software.ru/courses/lectures/2_Compilers2019.pdf
2. <https://www.intuit.ru/studies/courses/26/26/lecture/801?page=7>
3. Молчанов А.Ю. Системное программное обеспечение: учебник для вузов. 3-е изд. – СПб.: Питер, 2010. – 400 с. (С. 104-111)
4. А. Ахо, Дж. Ульман "Теория синтаксического анализа, перевода и компиляции", Т.1 "Синтаксический анализ", М.: Мир, 1978