Теория формальных языков и компиляторов

Часть 1. Порождающие грамматики и языки

Лекция 5. Языки порождающих грамматик

4.2.2 Прямая задача

Перейдём к прямой задаче, которая заключается в том, чтобы подобрать грамматику G, порождающую заданный язык L. Вновь обратимся к конкретным примерам.

Пример 4.5. Пусть: $L(G_5[Z]) = \{ a^n b^{2n-1} \mid n>=1 \}$. Необходимо построить грамматику $G_5[Z]$.

При подборе грамматики следует стремиться к тому, чтобы:

1) ограничить множество правил грамматики P;

2) ограничить множество нетерминальных символов V_N, по возможности избегая так называемых *цепных правил* (продукций).

Определение 4.8. Цепными называются продукции вида $A \to B$.

Грамматика считается оптимально сконструированной, когда при минимальных P и V_N порождается один и тот же язык. Для языка $L(G_5[Z])$ характерными являются цепочки, которые представляют собой n - кратную итерацию терминального символа a и (2 n – 1) – кратную итерацию символа b. По аналогии с примером 4.1 легко предложить $G_5[Z]$:

$V_T = \{a, b\}$,

$V_N = \{Z\}$,

P: $Z \to ab$

$Z \to aZbb$

Или ещё проще, пользуясь принятыми соглашениями^

P: $Z \to ab \mid aZbb$

Проверка правильности сконструированной грамматики осуществляется, например, уже рассмотренным методом выводимости. Действительно,

$$\begin{array}{ccccccc} Z & \Rightarrow & aZbb & \Rightarrow & aaZbbbb & \Rightarrow & \dots \\ & \Downarrow & & \Downarrow & & & \Downarrow \\ & ab & & aabbb & & & aaabbbbb \dots \end{array}$$

В следующем примере из класса прямых задач рассмотрим наиболее часто употребительные в инженерной практике решения задач – традиционные формы алгебраических выражений или как принято именовать эти конструкции в языковых процессорах – язык арифметических выражений (АВ).

Требуется решить прямую задачу разработки грамматики G[<АВ>] для записи арифметических выражений (формул) вида

a + в - c

a * (в / (c+d)) и т.д.

Эта нетривиальная задача, в отличие от показанных выше, решалась достаточно непросто. В её решении принимали участие коллективы отечественных и зарубежных учёных. Наиболее полно арифметические выражения представлены в языке FORTRAN. Приведём данную грамматику с сокращениями, которые не нарушают общности.

Усечение грамматики выполняется на уровне идентификаторов или операндов, в качестве которых будем использовать только терминальные символы a, b, c.

Пример 4.6. Пусть G[<АВ>] – усечённая (сокращенная, упрощенная) грамматика языка FORTRAN.

V_T={+, -, ∗, /, (,), a, b, c}

P: 1) <АВ> → Т (Т — терм)

2) <АВ> → Т + <АВ>

3) <АВ> → Т - <АВ>

4) Т → О (О — операнд)

5) Т → О∗Т

6) Т → О / Т

7) O → (<AB>) | a | b | c

Грамматика языка FORTRAN в отличие от других языков высокого уровня (C, C++, PASCAL, BASIC, JAVA) имеет операцию возведения в степень, которая в G[<AB>] не приведена.

Пример 4.7. Воспользуемся принятыми соглашениями и приведём грамматику к более компактному виду:

1. <AB> → T { < знак + > T }
2. < знак + > → + | -
3. T → O { < знак *> O }
4. < знак * > → * | /
5. O → (<AB>) | a | b | c

Выбор той или иной формы грамматики G[<AB>] связан с видом анализа текста, представляющего арифметические выражения. При синтаксическом анализе удобно пользоваться компактной формой. Например, при семантическом анализе, как будет показано ниже, можно пользоваться только безальтернативной расширенной формой.

В рассмотренных выше примерах решалась прямая или обратная задача поиска грамматики G по языку L или языка L по грамматике G.Теперь поставим задачу по-другому.

Пусть есть некоторая цепочка β, и требуется установить её принадлежность языку, порождаемому заданной грамматикой L(G[Z]), т.е. требуется установить, имеет ли место соотношение:

$$\beta \in L\left(G\left[Z\right]\right)?$$

Обратимся в очередной раз к примеру.

Пример 4.8. Пусть задана грамматика G_6[I] с множеством правил вывода P:

P: 1) I → abAI

2) I → c

3) A → bA

4) $A \to c$

И пусть требуется установить: принадлежит ли цепочка abbcc языку $L(G_6[I])$?

Воспользуемся всё тем же приёмом итерационной выводимости:

$$I \Rightarrow abAI \Rightarrow abAc \Rightarrow abbAc \Rightarrow abbcc \in \mathbf{L}(\mathbf{G_6}[I]).$$

Таким образом легко установлено, что данная цепочка принадлежит языку $L(G_6[I])$.

Во всех предыдущих примерах решения прямой и обратной задачи, а также задачи анализа решались с использованием определения выводимости. Это не единственный способ анализа. Существуют другие методы, например, построение синтаксических деревьев. Остановимся на этом более подробно в следующей лекции.

Упражнения

1. Постройте грамматики, порождающие следующие языки:

1.1. $L = \{1^n 0^m \mid n > 0, m > 0\}$

1.2. $L = \{1^n 0^n 1^m 0^m \mid n > 0, m > 0\}$

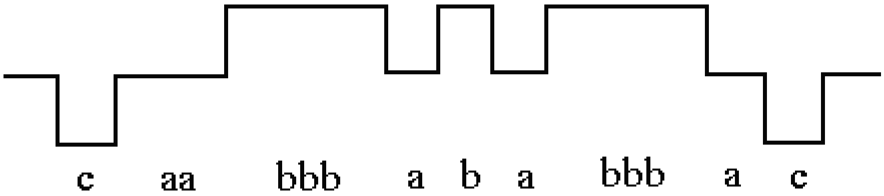
1.3. $L = \{a^m b^n c^k \mid m > 0, n > 0, \mathrm{k} > 0\}$

1.4. $L = \{a * + b *\}$

1.5. $L = \{a^n b^{3n-1} c^m \mid m \geq 0, n > 0\}$

1.6. $L = \{1^n 0^n \mid n > 0\} \cup \{a^{2n} b^n \mid n > 0\}$

2. Пусть задано множество биполярных сигналов потенциального типа, длительность которых изменяется дискретно. Начало и конец последовательности сигналов определяется сигналами отрицательной полярности.



Написать грамматику, задающую множество цепочек, соответствующих сигналам рассматриваемого типа, при условии, что состояния сигнала закодированы буквами a, b, c.

Список использованных источников

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