Лексикографический номер

Рассмотрим нумерацию, называемую лексикографической. В данной нумерации пустому слову присваивается номер 0, а буквам $a_1, \dots, a_n$ алфавита Σ — номера $1, \dots, n$ соответственно. Если слово x имеет лексикографический номер l_x, то слову xa_i присваивается номер $nl_x + i$. Отсюда следует, что лексикографический номер слова $a_{i_1}a_{i_2}\dots a_{i_k}$ будет равен

$$n^{k-1}{}_{i_1} + n^{k-2}{}_{i_2} + \dots + i_k.$$

Заметим, что последняя сумма напоминает запись числа в системе счисления по модулю n (мощности алфавита) с тем лишь различием, что используется цифра n, но не допускается цифра 0. Итак, по любому слову в алфавите Σ однозначно вычисляется его лексикографический номер. Обратно, любое натуральное число однозначно раскладывается по степеням n указанным выше образом.

Действительно, если дано число N, то при $0 \leqslant N \leqslant n$ оно служит номером пустого слова ($N = 0$) или некоторой буквы алфавита. Иначе представим N в виде

$$N = k_1 n + r_0,$$

где $1 \leqslant r_0 \leqslant n$.

Если $k_1 \leqslant n$, то N есть номер слова $a_{k_1}a_{r_0}$. Иначе раскладываем k_1 в виде

$$k_1 = k_2 n + r_1,$$

где $1 \leqslant r_1 \leqslant n$. Тогда

$$N = k_2 n^2 + r_1 n + r_0.$$

С числом k_2 поступаем точно так же, как и с k_1. После конечного числа шагов получим разложение числа N в виде

$$N = n^m r_m + n^{m-1} r_{m-1} + \dots + nr_1 + r_0,$$

где каждое число r_i ($0 \leqslant i \leqslant m$) находится в диапазоне от 1 до n.

По полученному разложению N однозначно восстанавливается слово в Σ, имеющее номер N.

Пример

Вычислим номер слова $cbaac$ в алфавите $\{a, b, c\}$. Имеем

$$3^4 \cdot 3 + 3^3 \cdot 2 + 3^2 \cdot 1 + 3^1 \cdot 1 + 3^0 \cdot 3 = 312.$$

Решим обратную задачу, найдя слово в данном трехбуквенном алфавите, имеющее номер 321.

Согласно приведенному выше алгоритму, получим

$$321 = 106 \cdot 3 + 3 = (35 \cdot 3 + 1) \cdot 3 + 3 = ((11 \cdot 3 + 2) \cdot 3 + 1) \cdot 3 + 3 =$$
$$= (((3 \cdot 3 + 2) \cdot 3 + 2) \cdot 3 + 1) \cdot 3 + 3 = ((3 \cdot 3^2 + 2 \cdot 3 + 2) \cdot 3 + 1) \cdot 3 + 3 =$$
$$= (3 \cdot 3^3 + 2 \cdot 3^2 + 2 \cdot 3 + 1) \cdot 3 + 3 = 3 \cdot 3^4 + 2 \cdot 3^3 + 2 \cdot 3^2 + 1 \cdot 3^1 + 3 \cdot 3^0.$$

Следовательно, искомое слово есть $cbbac$.

Введение

Говоря “формальный язык”, мы имеем в виду то, что приведенные в этом курсе результаты используются прежде всего при описании искусственных языков, придуманных людьми для специальных целей, например языков программирования. Но непреодолимой преграды между специально придуманными искусственными (формальными) языками и стихийно возникающими и развивающимися естественными языками не существует. Оказывается, что естественные языки характеризуются сложными грамматическими правилами, т.е. довольно жестко формализованы, а даже самый “научно разработанный” язык программирования содержит “темные места”, однозначное понимание которых является проблемой.

Изучая языки, следует иметь в виду три основных аспекта.

Первый из них — синтаксис языка. Язык — это какое-то множество "слов", где "слово" есть определенная конечная последовательность "букв" — символов какого-то заранее фиксированного алфавита. Термины "буква" и "слово" могут пониматься по-разному. Так, "буквами" могут быть действительно буквы алфавита какого-нибудь естественного или формального языка, например русского языка или языка программирования "Паскаль". Тогда "словами" будут конечные последовательности "букв": "крокодил", "integer". Такие слова называют "лексемами". Но "буквой" может быть "слово" ("лексема") в целом. Тогда "слова" — это предложения естественного языка или программы языка программирования. Если фиксировано какое-то множество "букв", то не каждая их последовательность будет "словом", т.е. "лексемой" данного языка, а только такая последовательность, которая подчиняется определенным правилам. Слово "крыкадил" не является лексемой русского языка, а слово "iff" не является лексемой в "Паскале". Предложение "Я люблю ты" не является правильным предложением русского языка, точно так же, как и запись "x :== t" не есть правильно написанный оператор присваивания "Паскаля". Синтаксис языка и представляет собой систему правил, в соответствии с которыми можно строить "правильные" последовательности "букв". Каждое слово языка характеризуется определенной структурой, специфичной именно для данного языка. Тогда необходимо, с одной стороны, разработать механизмы перечисления, или порождения, слов с заданной структурой, а с другой — механизмы проверки того, что данное слово принадлежит данному языку. Прежде всего именно эти механизмы и изучает классическая теория формальных языков.

Второй аспект — семантика языка. Семантика предполагает сопоставление словам языка некоего "смысла", "значения". Например, записывая математическую формулу, мы должны соблюдать определенные синтаксические правила (расстановка скобок, правописание символов, порядок символов и т.п.), но, кроме этого, формула имеет вполне определенный смысл, что-то обозначает.

Наконец, третий аспект — прагматика языка. Прагматика связана с теми целями, которые ставит перед собой носитель языка: например, человек произносит речь, имея перед собой цели, связанные не с синтаксисом, не с семантикой языка, на котором он говорит или пишет, а, скажем, с получением за речь определенной суммы денег. Прагматика является уже скорее дисциплиной социально-философской, затрагивающей целеполагающую деятельность личности.

Элементы теории формальных языков

Определение. Алфавит — это конечное множество символов.

Предполагается, что термин "символ" имеет достаточно ясный интуитивный смысл и не нуждается в дальнейшем уточнении.

Определение. Цепочкой символов в алфавите Σ называется любая конечная последовательность символов этого алфавита.

Определение. Цепочка, которая не содержит ни одного символа, называется пустой цепочкой. Для ее обозначения будем использовать греческую букву ε.

Предполагается, что сама буква ε в алфавит Σ не входит; она лишь помогает обозначить пустую последовательность символов.

Определение. Если α и β — цепочки, то цепочка $\alpha\beta$ (результат приписывания цепочки β в конец цепочки α) называется конкатенацией (или сцеплением) цепочек α и β. Конкатенацию можно считать двуместной операцией над цепочками: $\alpha \cdot \beta = \alpha\beta$.

Например, если $\alpha = ab$ и $\beta = cd$, то $\alpha \cdot \beta = abcd$.

Для любой цепочки α справедливы равенства: $\alpha \cdot \varepsilon = \varepsilon \cdot \alpha = \alpha$.

Для любых цепочек α, β, γ справедливо $(\alpha \cdot \beta) \cdot \gamma = \alpha \cdot (\beta \cdot \gamma)$ (свойство ассоциативности операции конкатенации).

Определение. Обращением (или реверсом) цепочки α называется цепочка, символы которой записаны в обратном порядке.

Обращение цепочки будем обозначать α^R.

Например, если $\alpha = abcdef$, то $\alpha^R = fedcba$.

Для пустой цепочки: $\varepsilon^R = \varepsilon$.

Определение. n-ой степенью цепочки α (будем обозначать α^n) называется конкатенация n цепочек α: $\alpha^n = \underbrace{\alpha\alpha\ldots\alpha\alpha\alpha}_{n}$.

Свойства степени: $\alpha^0 = \varepsilon$; $\alpha^n = \alpha\alpha^{n-1} = \alpha^{n-1}\alpha$.

Определение. Длина цепочки — это число составляющих ее символов (или длина последовательности символов).

Например, если $\alpha = abbcaad$, то длина α равна 7. Длину цепочки α будем обозначать $|\alpha|$. Длина ε равна 0.

Определение. Через $|\alpha|_s$ обозначают число вхождений символа s в цепочку α.

Например, $|babb|_a = 1$, $|babb|_b = 3$, $|babb|_c = 0$.

Определение. Обозначим через Σ^* множество, содержащее все цепочки в алфавите Σ, включая пустую цепочку ε.

Например, если $\Sigma = \{0, 1\}$, то $\Sigma^* = \{\varepsilon, 0, 1, 00, 11, 01, 10, 000, 001, 011, \ldots\}$.

Определение. Обозначим через Σ^+ множество, содержащее все цепочки в алфавите Σ, исключая пустую цепочку ε.

Следовательно, $\Sigma^* = \Sigma^+ \cup \{\varepsilon\}$.

Определение. Язык в алфавите Σ — это подмножество множества всех цепочек в этом алфавите. Для любого языка L справедливо $L \subseteq \Sigma^*$.

Известны различные способы описания языков. Конечный язык можно описать простым перечислением его цепочек. Поскольку формальный язык может быть и бесконечным, требуются механизмы, позволяющие конечным образом представлять бесконечные языки. Можно выделить два основных подхода для такого представления: механизм распознавания и механизм порождения (генерации).

Механизм распознавания (распознаватель), по сути, является процедурой специального вида, которая по заданной цепочке определяет, принадлежит ли она языку. Если принадлежит, то процедура останавливается с ответом «да», т.е. допускает цепочку; иначе — останавливается с ответом «нет» или зацикливается. Язык, определяемый распознавателем — это множество всех цепочек, которые он допускает.

Примеры распознавателей:

- Машина Тьюринга (МТ). Язык, который можно задать с помощью МТ, называется рекурсивно перечислимым. Вместо МТ можно использовать эквивалентные алгоритмические схемы: нормальный алгоритм Маркова (НАМ), машину Поста и др.
- Линейно ограниченный автомат (ЛОА). Представляет собой МТ, в которой лента не бесконечна, а ограничена длиной входного слова. Определяет контекстно-зависимые языки.
- Автомат с магазинной (внешней) памятью (МП-автомат). В отличие от ЛОА, головка не может изменять входное слово и не может сдвигаться влево; имеется дополнительная бесконечная память (магазин, или стек), работающая по дисциплине LIFO. Определяет контекстно-свободные языки.
- Конечный автомат (КА). Отличается от МП-автомата отсутствием магазина. Определяет регулярные языки.

Основной способ реализации механизма порождения — использование порождающих грамматик, которые иногда называют грамматиками Хомского.

Языки и операции над языками

Рассмотрим простые примеры языков в некоторым алфавите Σ:

1) $\varnothing$ — пустой язык;
2) $\{\varepsilon\}$ — язык, состоящий из пустой цепочки;
3) $\{a\}$, $a \in \Sigma$ — язык, состоящий из одной буквы a из алфавита Σ.

Определение. Суммой языков L_1 и L_2 называется язык, который обозначается L_1+L_2 и получается объединением множеств L_1 и L_2, т.е.

$$L_1 + L_2 = \{w \mid w \in L_1 \cup L_2\}.$$

Иными словами, сумма языков состоит из слов, принадлежащих хотя бы одному из языков L_1 и L_2. Поэтому $L_1 + L_2 = L_2 + L_1$.

Определение. Произведением языков L_1 и L_2 называется язык, который обозначается $L_1 \cdot L_2$ и получается в результате конкатенации всех возможных слов w_1 и w_2, где w_1 принадлежит языку L_1, а w_2 — языку L_2, т.е.

$$L_1 \cdot L_2 = \{w_1 w_2 \mid w_1 \in L_1,\ w_2 \in L_2\}.$$

Заметим, что язык $L_1 \cdot L_2$, как правило, отличается от языка $L_2 \cdot L_1$ (не выполняется условие коммутативности), хотя некоторые слова могут принадлежать обоим произведениям.

Пример

Пусть $L_1 = \{a, cb\}$, $L_2 = \{ab, bcb\}$. Тогда

$$L_1 \cdot L_2 = \{aab, abcb, cbab, cbbcb\},$$

$$L_2 \cdot L_1 = \{aba, abcb, bcba, bcbcb\}.$$

Слово $abcb$ принадлежит обоим языкам $L_1 \cdot L_2$ и $L_2 \cdot L_1$.

Далее будем использовать следующие обозначения: $L^0 = \{\varepsilon\}$, где ε — пустое слово, $L^1 = L$, $L^2 = L \cdot L$, $L^3 = L^2 \cdot L$, …, $L^k = L^{k-1} \cdot L$.

Определение. Итерацией языка L называется язык, который обозначается L^* и получается в результате сложения бесконечного числа языков $\{\varepsilon\}+L+L^2+L^3+\ldots+L^k+\ldots$, т.е.

$$L^* = \sum_{k=0}^{+\infty} L^k.$$

Итерация выражается через операции сложения и умножения языков. Из всех введенных операций над языками она единственная, которая позволяет из конечного языка получить бесконечный.

Пример

Найдем итерацию языков $L_1 = \{a\}$ и $L_2 = \{bc\}$. Согласно определению

$$L_1{}^* = \{\varepsilon\} + \{a\} + \{aa\} + \{aaa\} + \ldots = \{\varepsilon, a, aa, aaa, \ldots\} = \{a^n \mid n = 0, 1, 2, \ldots\},$$

$$L_2{}^* = \{\varepsilon\} + \{bc\} + \{bcbc\} + \ldots = \{\varepsilon, bc, bcbc, \ldots\} = \{(bc)^n \mid n = 0, 1, 2, \ldots\},$$

где $(a)^0 = (bc)^0 = \varepsilon$.

Иногда язык удобнее задавать не в виде множества, перечисляя слова, а с помощью выражений (формул), в которые входят слова и знаки операций сложения, умножения и итерации. Например, для языков $L_1 = \{a, cb\}$ и $L_2 = \{ab, bcb\}$ можно использовать равенства $L_1 = a + cb$, $L_2 = ab + bcb$. Тогда

$$L_1 \cdot L_2 = (a+cb)\cdot(ab+bcb) = a\cdot ab + a\cdot bcb + cb\cdot ab + cb\cdot bcb = aab + abcb + cbab + cbbcb.$$

Итерацию языков $L_1 = \{a\}$ и $L_2 = \{bc\}$ можно записать следующим образом:

$$L_1{}^* = \varepsilon + a + a^2 + a^3 + \ldots + a^n + \ldots = a^*,$$

$$L_2{}^* = \varepsilon + bc + (bc)^2 + \ldots + (bc)^n + \ldots = (bc)^*.$$

Таким образом, с помощью введенных операций сложения, умножения и итерации некоторые языки можно выражать в виде формул через более простые языки. Причем результатом сложения или умножения двух конечных языков всегда будет конечный язык, и лишь итерация позволяет из конечного языка получить бесконечный. Отметим основные свойства свойства операций над языками:

1) $L_1 \cdot (L_2 + L_3) = L_1 \cdot L_2 + L_1 \cdot L_3$;
2) $(L_1 + L_2) \cdot L_3 = L_1 \cdot L_3 + L_2 \cdot L_3$;
3) $L + L = L$;
4) $L + L^* = L^*$;
5) $L \cdot \varepsilon = L$;
6) $L \cdot L^* = L^* \cdot L$;
7) $\varepsilon + L \cdot L^* = L^*$;
8) $({L_1}^* \cdot {L_2}^*)^* = (L_1 + L_2)^*$.

Регулярные языки и регулярные выражения

Очевидно, что для любого языка L верны равенства $L + \varnothing = L$ и $L \cdot \varnothing = \varnothing$. Значит, при всех натуральных значениях n выполняется $\varnothing^n = \varnothing$. Тогда из определения итерации получаем

$$\varnothing^* = \varepsilon + \varnothing + \varnothing^2 + \varnothing^3 + \ldots + \varnothing^n + \ldots = \varepsilon.$$

Заметим также, что $\varepsilon^* = \varepsilon$, поскольку $\varepsilon^n = \varepsilon$ и $\varepsilon + \varepsilon = \varepsilon$.

Определение. Пусть имеется алфавит $A = \{a_1, a_2, \ldots, a_s\}$. Одноэлементные языки $a_1, a_2, \ldots, a_s$, а также язык, содержащий только пустое слово ε, будем называть элементарными языками.

Определение. Регулярным языком называется такой язык, который можно получить из элементарных языков с помощью конечного числа операций сложения, умножения и итерации.

Чтобы доказать регулярность какого-либо языка, надо записать его в виде так называемого регулярного выражения, т.е. формулы, в которой конечное число раз используются элементарные языки и знаки операций сложения, умножения и итерации. Поскольку количество регулярных выражений счетно, то число различных регулярных языков не более, чем счетно. Всего же имеется континуум языков над фиксированным конечным алфавитом , т.к. язык — это любое подмножество счетного множества *. Следовательно, существуют и нерегулярные языки.

Пример

Рассмотрим несколько языков.

1) Конечный язык $L_1 = \{a, ab, abc\}$ является регулярным языком, т.к. его можно задать равенством $L_1 = a + ab + abc = a + a \cdot b + a \cdot b \cdot c = a \cdot (\varepsilon + b \cdot (\varepsilon + c))$. Последнее полученное выражение является регулярным, поскольку оно содержит только простейшие языки a, b, c и ε и конечное число знаков операций сложения и умножения. Этот пример показывает, что любое конечное множество слов образует регулярный язык.
2) Бесконечный язык $L_2 = \{c, cabc, cabcabc, cabcabcabc, \ldots\}$ является регулярным, т.к. его можно задать разными регулярными выражениями: $c \cdot (a \cdot b \cdot c)^*$, либо $(c \cdot a \cdot b)^* \cdot c$. Этот пример свидетельствует о том, что один и тот же язык можно представить через различные регулярные выражения.
3) Бесконечный язык L_3, состоящий из всех слов конечной длины в алфавите $A = \{a, b, c\}$, включая и пустое слово, является регулярным языком, поскольку выполняется равенство $L_3 = (a + b + c)^*$.
4) Бесконечный язык L_4 над алфавитом $A = \{a, b, c\}$, образованный словами, которые содержат хотя бы одну букву c, регулярен, т.к. он может быть задан равенством $L_4 = (a + b + c)^* \cdot c \cdot (a + b + c)^*$.
5) Бесконечный язык L_5 над алфавитом $A = \{0, 1\}$, образованный всеми словами, кроме слов 0 и 11, регулярен, т.к. его можно задать регулярным выражением