

УДК 510

ВИКОРИСТАННЯ ТЕОРІЇ ФОРМАЛЬНИХ МОВ І ГРАМАТИК У МАТЕМАТИЧНІЙ ЛОГІЦІ

В. М. Євладенко

У статті розглядається застосування теорії формальних мов і граматики у математичній логіці.

The paper views the application of the theory of formal languages and grammars in mathematical logics.

Основними розділами, які вивчаються у вузівському курсі математичної логіки є: алгебра висловлювань, числення висловлювань, логіка предикатів, числення предикатів та математичні теорії.

При цьому алгебра висловлювань та логіка предикатів вивчаються на змістовному рівні. Числення висловлювань, як і числення предикатів, вивчаються строго математично (аксіоматично). З цією метою в кожному із цих числень фіксується формальна мова, в якій дається строге означення формули і поняття вивідних формул (змістовно-тавтологій). Як поняття формули, так і поняття вивідної формули носять рекурсивний характер, а саме: задаються (перераховуються) початкові вивідні формули (аксіоми) і задаються формально правила виводу, які дозволяють із відомих вивідних формул одержувати нові вивідні формули. Очевидно, що на першому етапі вивідними є тільки аксіоми відповідного числення.

Отже, як поняття формул, так і поняття вивідних формул в цих логічних численнях носять чисто формальний і в той же час строго математичний характер, а тому вони досить легко піддаються алгоритмізації і відповідній комп'ютерній реалізації. В цьому плані рекомендуємо використовувати основи теорії формальних мов і граматики [1].

Алфавітом називають не порожню скінченну множину символів (букв) x_i , які чітко відрізняються один від одного: $T = \{x_1, x_2, \dots, x_n\}$. Під словом (ланцюгом) $x_{i_1}, x_{i_2}, \dots, x_{i_k}$ довжини $k \geq 0$ в алфавіті T розуміють будь-яку скінченну послідовність букв $x_i \in T$ ($i = 1, \dots, n$). Слово довжиною 0 називають порожнім і позначають буквою ε . Множину всіх слів, включаючи і порожнє слово ε , в алфавіті T називають універсальною мовою (так звана вільна півгрупа) і позначають T^* . Формальна мова L в алфавіті T – це деяка підмножина множини T^* , тобто $L \subseteq T^*$. Мова L може бути скінченною чи нескінченною. У випадку, коли L – скінченна, тобто містить скінченну множину слів, її можна задати простим перерахунком всіх її слів. У разі, коли L – нескінченна, вказаним способом (тобто перерахунком слів) її в принципі неможливо задати. В цьому випадку розробляється скінченне число правил (строго математичний апарат), які дають можливість конструктивно задати нескінченну мову. Одним з таких апаратів є породжуючі граматики Н.Хомського [1].

Породжуючою граматиною називається впорядкована четвірка $G = \langle T, N, P, y_1 \rangle$, де $T = \{x_1, x_2, \dots, x_n\}$ – термінальний алфавіт (алфавіт основних символів); $N = \{y_1, y_2, \dots, y_s\}$ – нетермінальний алфавіт (алфавіт метасимволів або допоміжних символів); P – скінченна множина впорядкованих пар (u, v) , де $u, v \in T^*$, які називають підстановками (продукціями) і записують у вигляді $u \rightarrow v$; y_1 – так звана аксіома ($y_1 \in N$, y_1 – початковий метасимвол).

Слово w_{i+1} називають безпосередньо вивідним із слова w_i і пишуть $w_i \Rightarrow w_{i+1}$, якщо існують слова α_1, α_2, u, v такі, що $w_i = \alpha_1 u \alpha_2$,

$w_{i+1} = \alpha_1 v \alpha_2$ і $u \rightarrow v \in P$. Далі, слово w_r називають вивідним із слова w_0 (позначення $w_0 \Rightarrow w_r$) за r кроків виводу, якщо існують w_i такі, що

$w_i \Rightarrow w_{i+1}$ для всіх i . Послідовність w_0, w_1, \dots, w_r називають виводом довжини r і записують $w_0 \Rightarrow w_1 \Rightarrow w_2 \Rightarrow \dots \Rightarrow w_r$.

Множина $L(G) = \{x \in T^* / y_1 \Rightarrow x\}$ називається мовою, породженою граматиною G . Іншими словами, мова – це множина слів в термінальному алфавіті, вивідних із аксіоми y_1 .

Накладаючи обмеження на структуру підстановок $u \rightarrow v$ серед породжуючих граматинок виділяють чотири основних підкласи – типу 0, 1, 2, 3.

Клас граматинок типу 0 – це породжуючі граматики загального типу, на підстановки яких ніяких обмежень не накладається.

Граматики типу 1 (так звані контекстні) – це породжуючі граматики, підстановки яких мають структуру: $\alpha_i y_i \beta_i \rightarrow \alpha_i w \beta_i$, тобто метасимвол $y_i \in N$ замінюється на слово $w \in (T \cup N)^*$.

Клас граматинок типу 2 – контекстно-вільні (к-в) граматики. В них підстановки мають вид $y_i \rightarrow w_i$ ($y_i \in N$, $w_i \in (T \cup N)^*$), тобто метасимвол y_i замінюється на слово w_i , незважаючи на контекст.

Граматики типу 3 – автоматні граматики, підстановки яких мають наступну структуру: $y_i \rightarrow x_j, y_i \rightarrow x_j y_k$ ($x_j \in T, y_i, y_k \in N$).

Мова, породжена граматиною типу k ($k = 0, 1, 2, 3$) називається мовою типу k .

Із чотирьох виділених класів породжуючих граматинок широке застосування в області програмування мають к-в граматики. Це пов'язано з тим, що синтаксис сучасних мов програмування (алголоподобних мов) досить просто описується так званими бекусівськими нормальними формами (БНФ), які еквівалентні контекстно-вільним граматикам. У БНФ система підстановок P записується у компактному вигляді, наприклад:

$$y_1 ::= a y_2 y_3 \mid y_2 b y_3$$

$$y_2 ::= a | b | c y_2$$

$$y_3 ::= c | b y_3$$

де символ “ $::=$ ” означає заміну, а символ “ $|$ ” читається як “або”, тобто як і в к-в граматиці кожний метасимвол y_i можна замінити на одне із альтернативних слів, записаних в правій частині від символа “ $::=$ ”.

Тепер, користуючись апаратом к-в граматик, можна задати синтаксис мови формул числення висловлювань, наприклад, наступним чином.

Термінальний алфавіт (основні символи), в якому записуються формули, складається із символів трьох категорій:

а) великі латинські букви A, B, C, \dots, X, Y, Z – так званні змінні висловлювання, та дві константи – 0 і I .

б) символи логічних операцій: \neg (заперечення), \wedge (кон’юнкція), \vee (диз’юнкція), \rightarrow (імплікація);

в) відкриваюча та закриваюча круглі дужки: $(,)$.

Формула числення висловлень – це деяка скінченна послідовність букв термінального алфавіту, але не будь-яка така послідовність буде формулою. Точне означення формули носить рекурсивний характер [2]:

1) змінні висловлювання A, B, C, \dots, X, Y, Z та константи 0 і I – формули;

2) якщо U_1 і U_2 – формули, то формулами будуть і слова: $(\neg U_1)$, $(\neg U_2)$, $(U_1 \wedge U_2)$, $(U_1 \vee U_2)$, $(U_1 \rightarrow U_2)$.

Тоді к-в граматика G_1 , яка породжує формули числення висловлювань, – це впорядкована четвірка виду $G_1 = \langle T_1, N_1, P_1, y_1 \rangle$, де $T_1 = \{A, B, \dots, Y, Z\}$ – термінальний алфавіт, N_1 – алфавіт понять (мета-символів), число яких визначиться пізніше, виходячи із системи підстановок P_1 , які задаються такою БНФ:

$$\langle \text{формула} \rangle ::= \langle \text{елементарна формула} \rangle | (\langle \neg \text{елементарна формула} \rangle) |$$

$$| (\neg \langle \text{формула} \rangle) | (\langle \text{формула} \rangle \langle \text{двомісна логічна операція} \rangle \langle \text{формула} \rangle)$$

$$\langle \text{елементарна формула} \rangle ::= \langle \text{змінне висловлення} \rangle | \langle \text{постійне висловлення} \rangle$$

$$\langle \text{змінне висловлення} \rangle ::= A | B | \dots | X | Y | Z$$

$$\langle \text{постійне висловлення} \rangle ::= 0 | I$$

$$\langle \text{двомісна логічна операція} \rangle ::= \wedge | \vee | \rightarrow$$

Якщо замість змістовних понять введемо формально відповідні метасимволи, використавши, наприклад, такі позначення:

$$\langle \text{формула} \rangle = y_1$$

$$\langle \text{елементарна формула} \rangle = y_2$$

$$\langle \text{змінне висловлення} \rangle = y_3$$

$$\langle \text{постійне висловлення} \rangle = y_4$$

$$\langle \text{двомісна логічна операція} \rangle = y_5,$$

то к-в граматика G_1 буде задаватися такою БНФ:

$$\begin{aligned}
 y_1 &::= y_2 \mid (\neg y_2) \mid (\neg y_1) \mid (y_1 y_5 y_1) \\
 y_2 &::= y_3 \mid y_4 \\
 y_3 &::= A \mid B \mid \dots \mid Y \mid Z \\
 y_4 &::= 0 \mid I \\
 y_5 &::= \wedge \mid \vee \mid \rightarrow
 \end{aligned}$$

Отже, граматику $G_1 = \langle T_1, N_1, P_1, y_1 \rangle$, де $N_1 = \{ y_1, y_2, y_3, y_4, y_5 \}$, а y_1 – аксіома.

Мова $L_1(G_1) = \{ \alpha \in T^* / y_1 \Rightarrow \alpha \}$ – це множина всіх формул числення висловлень. Наприклад, формула $\alpha = ((\neg A) \vee ((B \wedge C) \rightarrow (\neg B)))$ одержується в результаті такого виводу: $y_1 \Rightarrow (y_1 y_5 y_1) \Rightarrow ((\neg y_2) y_5 y_1) \Rightarrow ((\neg y_3) y_5 y_1) \Rightarrow ((\neg A) y_5 y_1) \Rightarrow ((\neg A) \vee y_1) \Rightarrow ((\neg A) \vee (y_1 y_5 y_1)) \Rightarrow ((\neg A) \vee ((y_1 y_5 y_1) y_5 y_1)) \Rightarrow \dots \Rightarrow ((\neg A) \vee ((y_2 y_5 y_1) y_5 y_1)) \Rightarrow ((\neg A) \vee ((y_3 y_5 y_1) y_5 y_1)) \Rightarrow ((\neg A) \vee (B y_5 y_1) y_5 y_1) \Rightarrow \dots \Rightarrow ((\neg A) \vee (B \wedge C) \rightarrow (\neg y_2)) \Rightarrow ((\neg A) \vee (B \wedge C) \rightarrow ((\neg y_3))) \Rightarrow ((\neg A) \vee ((B \wedge C) \rightarrow (\neg B))) = \alpha$.

Отже, $y_1 \Rightarrow \alpha$.

Тепер є можливість автоматично одержувати слова, які є формулами числення висловлень. Для цього досить розробити комп'ютерну програму, яка за скінченне число кроків виводу дає ту чи іншу формулу. При цьому, якщо в правих частинах БНФ для y_i маємо k ($k > 1$) різних альтернативних підстановок, то в процесі автоматичного виводу буде застосовуватися тільки перша підстановка. Це може привести до того, що множина так вивідних формул дає мову $L_2 \subset L_1(G_1)$. Крім того, в силу рекурсивності підстановок можливе зациклювання на метасимволах, що веде до нескінченності процесу виводу. Ці колізії усуваються шляхом резервування можливості чисто випадково застосовувати будь-яку з k альтернативних підстановок для кожного y_i . Для цього доцільно використати датчик випадкових чисел від 1 до k , який на кожному кроці виводу буде вказувати номер альтернативної підстановки для кожного y_i . Оскільки процес виводу будь-якої формули повинен бути скінченним, то після r кроків виводу (де r може бути досить великим) у вивідному слові w_i всі метасимволи треба остаточно замінити на термінальні слова.

Однією з важливих проблем, яку тепер легко розв'язати, є проблема так званого синтаксичного контролю формул числення висловлювань. Проблема автоматичного розпізнавання синтаксичної правильності формул полягає в наступному. Є k -в граматику G_1 , що задає мову $L_1(G_1)$ формул числення висловлювань, і маємо текст α в алфавіті T_1 цієї мови. Вияснити,

$$\alpha \in L_1(G_1)?$$

При розробці блоків синтаксичного контролю сучасних систем програмування використовуються два методи контролю [3]: «розгорткою»

або «зверху-вниз» (від аксіоми y_1 до слова α) та «згорткою» або «знизу-вверх» (від слова α до аксіоми y_1).

Ідея алгоритму синтаксичного контролю методом «розгортки» така. Використовуємо два стеки – вхідний та стек виходу. На початку роботи у вхідний стек поміщаємо контрольований текст α , а в стек виводу – аксіому y_1 . Далі із аксіоми y_1 прагнемо посимвольно вивести слово α . Відповідний алгоритм контролю, орієнтований на будь-яку k -в граматику, буде багатоперебірним. Це означає, що якщо в процесі одного з можливих виводів ми не одержали слова α , то потрібно перебирати всі інші можливі виводи (у відповідності з деревом виводу). Якщо жоден із таких виводів не дає слова α , то $\alpha \notin L_1(G_1)$, тобто α не є формулою числення висловлювань. Якщо ж один із виводів дає α , то $\alpha \in L_1(G_1)$.

Отже, використовуючи апарат формальних мов і граматик, розроблений в інформатиці, можна розв'язувати такі задачі числення висловлювань:

- 1) автоматично одержувати тексти, що є формулами числення висловлювань;
- 2) вирішувати проблему синтаксичного контролю для формул цього числення.

Далі можна формалізувати та алгоритмізувати поняття вивідної формули в численні висловлювань, задавши аксіоми та правила виводу, і автоматично одержувати вивідні формули.

Висвітлені в статті проблеми встановлюють тісний взаємозв'язок вузівських курсів інформатики та математичної логіки.

БІБЛІОГРАФІЯ

1. Гинзбург С. Математическая теория контекстно-свободных языков. –М.: Мир, 1970.
2. Новиков П.С. Элементы математической логики. -М.: Наука, 1976.
3. Братчиков Н.Л. Синтаксис языков программирования. –Новосибирск: Наука, 1975.

УДК 51(09)

ВИТОКИ СУЧАСНОЇ НУМЕРОЛОГІЇ

Т.В. Жабо

У статті розглядається історія розвитку чисел від шумерської нумерології до сучасних арабських цифр.

There is a history about number's development from shumer numerology till modern arabic figures in the article.

Числа є комбінаторними конфігураціями цифр, які в свою чергу є ідеографічними символами сучасної писемної мови. Ще й зараз наука не може відповісти на запитання, коли і як виникли перші цифри. Майже всі числові системи будувались за десятковим принципом. Чи не єдиним