

3.2. ОСНОВНІ ТЕОРЕТИЧНІ ВІДОМОСТІ З ФОРМАЛЬНИХ ГРАМАТИК. НОРМАЛЬНІ ФОРМИ БЕКУСА-НАУРА

Формально граматика визначається четвіркою $G=(V_n, V_t, S, P)$, де:
 V_n і V_t – множини *нетермінальних*, що не перетинаються, і *термінальних символів* відповідно;

S – виділений символ у V_n , що звичайно називають вихідним або початковим символом;

P – кінцева множина продукцій (правил), за якими нетермінальні символи визначаються як упорядкована послідовність термінальних та/або нетермінальних символів. Ця послідовність може складатися з одного термінального або нетермінального символу. В останньому випадку це буде просте перевизначення нетермінального символу лівої частини правила.

Об'єднання множин V_n і V_t називають *словником* граматики.

Для завдання продукцій частіше всього використовують метамова (мова для опису мов) нормальних форм Бекуса-Наура або Бекусових нормальних форм (скорочено – БНФ), яка була запропонована і вжита у 1955 р. для опису однієї з перших і на той час найбільш поширеної універсальної мови програмування АЛГОЛ. У цій метамові використовуються наступні метасинтаксичні символи або сполучення символів:

<> – кутні дужки для визначення в них імені нетермінального символу;

::= – аналог оператора присвоювання в мовах програмування. Він читається як «це є» і розділяє ліву і праву частини граматичного правила.

| – роздільник альтернативних правил у правій частині металінгвістичної формули.

У лівій частині формули завжди повинний бути присутнім нетермінальний символ. Якщо в лівій частині будь-якої формули якоїсь граматики є не більш одного нетермінального символу, то така граматика є *контекстно незалежною*, у протилежному випадку – *контекстно залежною*. Далі будемо розглядати тільки *контекстно незалежні* граматики.

У правій частині формули записується одне або декілька (якщо вони існують) правил, розділених символом « \gg », кожне з яких являє собою, як наводилося, послідовність термінальних та/або нетермінальних символів.

Кожний нетермінальний символ повинний зустрітися в лівій частині якогось правила, тобто не може бути жодного невизначеного нетермінального символа.

Наведемо приклад граматики такого примітивного оператора присвоєння змінній значення арифметичного виразу: він складається з двох операндів і однієї з арифметичних операцій $\{+, -, *, /\}$, при цьому дужки не використовуються, іменем змінної може бути тільки прописна літера латинського алфавіту із множини $\{a, b, c, d, e, f\}$, операндами арифметичного виразу може бути змінна, ім'я якої складається з однієї літери з множини $\{a, b, c, d, e, f\}$ або цифра.

Грамматика примітивного арифметичного оператора (G1)

- (1) $\langle \text{оператор_присвоєння} \rangle ::= \langle \text{змінна} \rangle := \langle \text{вираз} \rangle$
- (2) $\langle \text{змінна} \rangle ::= \langle \text{буква} \rangle$
- (3) $\langle \text{вираз} \rangle ::= \langle \text{терм} \rangle \langle \text{знак операції} \rangle \langle \text{терм} \rangle$
- (4, 5) $\langle \text{терм} \rangle ::= \langle \text{змінна} \rangle \mid \langle \text{число} \rangle$
- (6) $\langle \text{число} \rangle ::= \langle \text{цифра} \rangle$
- (6-9) $\langle \text{знак операції} \rangle ::= + \mid - \mid * \mid /$
- (10-15) $\langle \text{буква} \rangle ::= a \mid b \mid c \mid d \mid e \mid f$
- (16-25) $\langle \text{цифра} \rangle ::= 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$

Примітка. В дужках наведено номери правил граматики.

Граматику *G1*, що представлена вище у БНФ, можна представити і в графічній формі, що робить її більш наочною. Вона представлена на рис. 3.1.

На рисунку вживана така нотація. Прямокутниками представлені нетермінальні символи, кружками – термінальні. Кожній вертикальній лінії відповідає синтаксичне правило (продукція), а горизонтальні направлені лінії розкривають послідовність термінальних та/або нетермінальних символів, складаючи таким чином сутність правила.

Як бачимо, крім обмеженості, обумовленої алфавітом, в заданій граматиці є ще обмеження на кількість знаків у імені змінної й у числа (по одному знаку), а також на кількість операндів (всього два) і операторів (всього один) у виразі. На перший погляд, для зняття цих обмежень можна обійтися простим збільшенням складових у правилах, наприклад:

$\langle \text{число} \rangle ::= \langle \text{цифра} \rangle \mid \langle \text{цифра} \rangle \langle \text{цифра} \rangle ,$

тоді можна вже використовувати двозначні цілі числа. Але існує більш універсальний механізм розв'язання завдання розмірності граматичних конструкцій – рекурсія. У загальному випадку рекурсію можна розглядати як процес, у якій щось визначається саме через себе.

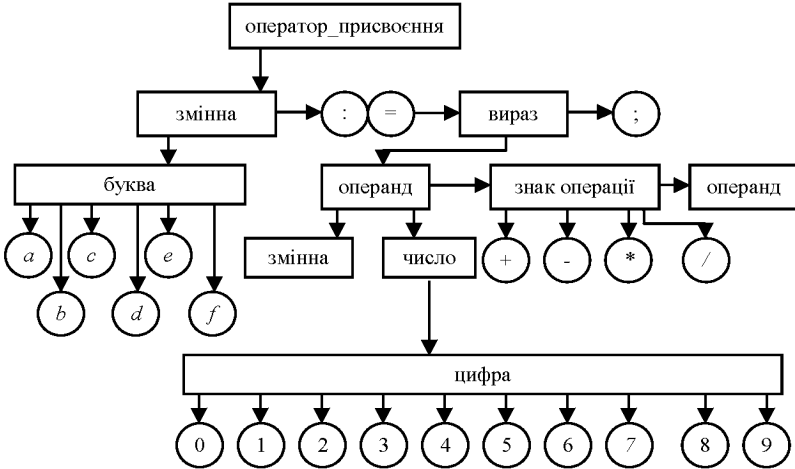


Рис. 3.1. Графічне представлення граматики $G1$

Нижче наводиться приклад граматики, що рекурсивно визначає просту мову визначення ідентифікаторів як сукупності літер і цифр, що починається з літери:

Грамматика визначення ідентифікатора ($G2$)
 $\langle \text{ідентифікатор} \rangle ::= \langle \text{літера} \rangle | \langle \text{ідентифікатор} \rangle \langle \text{літера} \rangle$
 $| \langle \text{ідентифікатор} \rangle \langle \text{цифра} \rangle$
 $\langle \text{літера} \rangle ::= a | b | \dots | z | _$
 $\langle \text{цифра} \rangle ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9$

Графічно ця грамматика представлена на рис. 3.2.

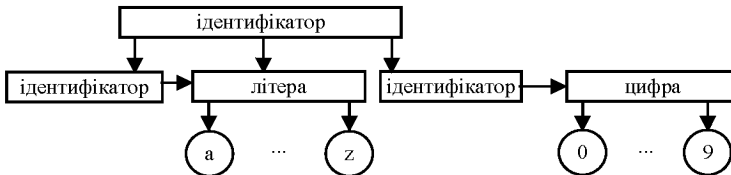


Рис. 3.2. Грамматика $G2$

Не всі рядки, утворені з термінальних символів, являють собою синтаксично правильні «речення» цієї мови, наприклад, «a», «a1», «ghb4r» – це речення даної мови, а «4», «ADE» – не є реченнями цієї мови.

Аналіз сполучень різноманітних елементів рядка для визначення того, є чи не є даний рядок символом реченням мови, називається граматичним розбором. Для синтаксично правильного речення можна побудувати дерево граматичного розбору, у той час як для рядка, що не є реченням даної мови, такої схеми не існує. Зобразимо дерево граматичного розбору для рядка «fs8» мовою, граматику якої була наведена вище (рис. 3.3).

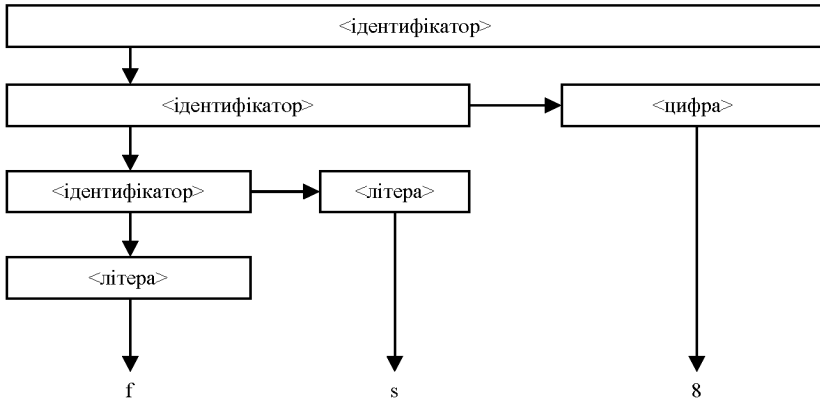


Рис. 3.3. Дерево граматичного розбору рядка «fs8»

Наведемо ще один приклад граматики частини мови Pascal з урахуванням того, що граматику нетермінального символу «ідентифікатор» було наведено вище (див. G2). Обмежимося також трьома типами даних.

Граматику оператора визначення змінних у мові Pascal (G3)

```

<опер_опису_змінних> ::= var <список_імен_змінних> :
    <тип_змінних>;
<список_імен_змінних> ::= <ім'я_змінної> | <список_імен_
    змінних>, <ім'я_змінної> |
<ім'я_змінної> ::= <ідентифікатор>
<тип_змінних> ::= char | real | int
    
```

Наведеній граматиці відповідають, наприклад, такі рядки:

```

var ac2: char;
var bb: int;
var c2,a,b1d: real;
    
```