

Лекція №13

Семантика контекстно-вільних мов і генерація вихідного коду

Способи визначення семантики мов програмування

Розрізняють формальну і неформальну семантику.

Формальна семантика (для побудови компіляторів) – це зміст (сенси) синтаксичних правил вхідної мови, що виражені в термінах вихідної мови та в діях по формуванню значень вихідної мови.

Неформальна семантика (для користувача) – це зміст (сенси) речень вхідної мови, що виражений в термінах речень вихідної мови.

В якості метасемантичної мови може бути або спеціалізована метасемантична мова, або інша мова програмування. Серед відомих метасемантичних мов можна назвати:

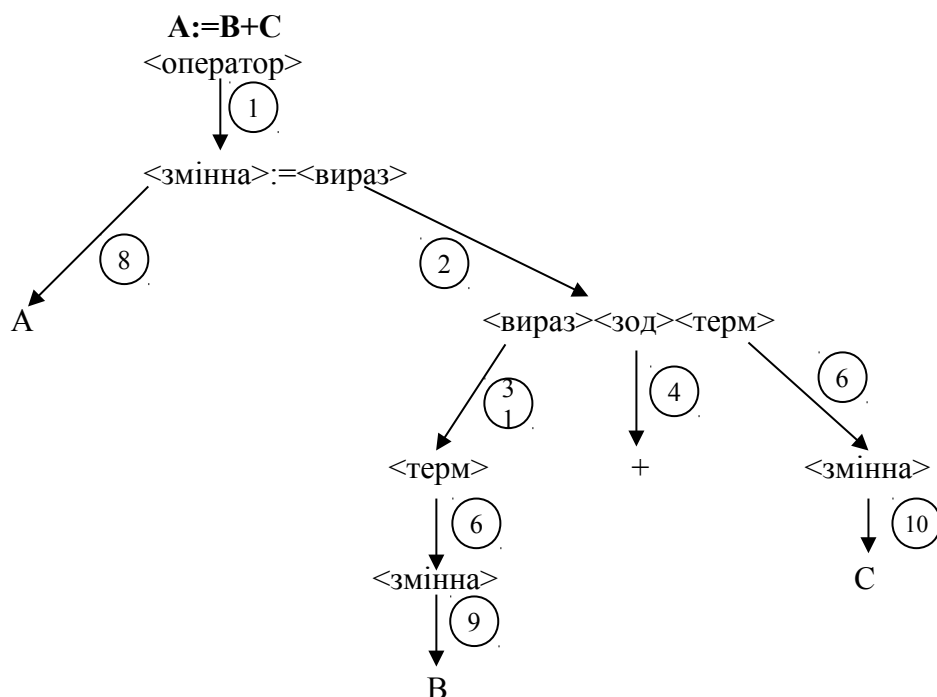
- 1) дворівневі або w-граматики;
 - 2) системи продукцій;
 - 3) віденську метамову;
 - 4) атрибутні граматики (розширений атрибутний метод).
-
1. Дворівневі або w-граматики використовувалися для опису семантики мови АЛГОЛ-68 на основі використання метаправил, які дозволяють описати контекстно-залежні характеристики мови у вигляді правил КВ-граматики.
 2. Система продукцій дозволяє сформувати породжуюче правило, яке визначає семантику на основі контекстно-залежних характеристик мови.
 3. Віденська метамова дозволяє описати процедуру, що породжує речення на деякій об'єктній мові абстрактної машини, інтерпретуючи яку можна згенерувати код для конкретного комп'ютера. Така метамова використовувалася для реалізації компілятора мови PL-1.
 4. Метод атрибутних граматик оснований на тому, що правила КВ-мови доповнюються атрибутами, здійснивши обчислення яких можна отримати вихідну програму. Атрибути приписуються нетерміналам і можуть бути *успадкованими* (спадкування атрибутів відбувається зверху донизу) і *синтезованими* (синтез атрибутів відбувається знизу вгору).

Внутрішнє подання дерева розбору

Синтаксис:

1. <оператор> → <змінна>:=<вираз>
2. <вираз> → <вираз><зод><терм>
3. <вираз> → <терм>
4. <зод> → +
5. <зод> → –
6. <терм> → <змінна>
7. <терм> → (<вираз>)
8. <змінна> → A
9. <змінна> → B
10. <змінна> → C
11. <змінна> → D

При побудові дерева розбору розрізняють прості вузли, що складаються із одного нетерміналу, і складні, що складаються із декількох компонент.



У внутрішньому поданні дерево розбору часто подається в вигляді цілочисельного вектора. Елементи такого вектора мають наступний зміст:

- додатне ціле число є номером правила;
- від’ємне ціле число є посиланням (індексом елемента вектора, в якому розміщується номер потрібного правила).

Подання дерева розбору у вигляді числового вектора називається **лінійною формою подання дерева розбору**.

Порядок побудови лінійної форми з посиланнями

1. Дерево розбору проходять зліва-направо, знизу-вверх по ланцюжкам, які ведуть до нелівих нетерміналів.
2. При досягненні нелівого нетерміналу в вектор розбору (назвемо його вектором RAS) виписуються номери правил, що йдуть зверху-вниз. Якщо інших нелівих нетерміналів у даному правилі більше немає, то подальший підйом вверх можливий тільки в тому випадку, коли шлях веде до нелівої компоненти або аксіому граматики.
3. Якщо складний вузол містить декілька нелівих компонент, то кожній нелівій компоненті ставиться у відповідність посилання, причому порядок посилань нумерується справа наліво.
4. Останнім елементом вектора RAS є посилання на аксіому граматики.

LRAS												
1	2	3	4	5	6	7	8	9	10	11	12	13
4	6	10	2	-2	-1	3	6	9	1	-4	8	-10

Довжина вектора RAS для розглянутого вище дерева дорівнює 13-ти елементам (LRAS=13).

Види семантичної відповідності

1. Проста відповідність – на вихід передається один символ або рядок символів вихідної мови.
2. Відповідність типу посилання (клауза) – рекурсивний спуск по дереву розбору.
3. Проста функціональна відповідність – значення або спосіб обробки нетерміналу є функцією його входження в праву частину правила.

Приклад.

Для нетерміналу <зод> при обробці правил

<множник> → <зод><число>

та

<вираз> → <вираз><зод><терм>

повинні бути згенеровані різні команди вихідної мови.

4. Складна функціональна відповідність – семантика обробки нетерміналу визначається дальнім контекстом.

Приклад.

В правилі <змінна>:=<вираз>, змінна і вираз можуть бути різних типів (дійсних, цілих, тощо) і, відповідно до цих типів, повинні бути згенеровані різні команди вихідної програми.

Проста метасемантична мова

Проста метасемантична мова – це множина семантичних визначень, які описуються трьома простими видами об'єктів і які ставляться у відповідність кожному синтаксичному правилу.

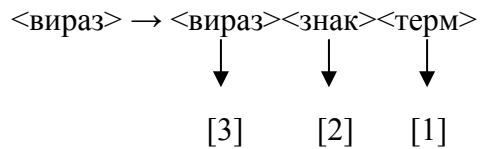
Семантичне визначення – це опис відповідності між кожним правилом (іноді конструкцією) вхідної мови та конструкцією (-ями) (фрагментом коду) вихідної мови, а також опис дій, що виконують семантичний аналіз та генерацію коду, для даного правила граматики.

Розглянемо граматику простої метасемантичної мови.

1. <семантичне визначення> → {<сукупність об'єктів>}
2. <сукупність об'єктів> → <об'єкт>
3. <сукупність об'єктів> → <об'єкт><сукупність об'єктів>
4. <об'єкт> → <простий об'єкт>
5. <об'єкт> → <посилання>
6. <об'єкт> → <розділювач>
7. <простий об'єкт> → <рядок символів>
8. <рядок символів> → <символ>
9. <рядок символів> → <символ><рядок символів>
10. <посилання> → [<клауза>]
11. <клауза> → <ціле>
12. <ціле> → <номер нетермінальної компоненти правої частини правила, рахуючи справа наліво>
13. <розділювач> → _
14. <розділювач> → α

Порядок слідування клауз (посилань для спуску по дереву) в семантичному визначенні задає порядок розбору.

Приклад:



Семантичне визначення для даного правила може бути, наприклад, таким:
{[3][2][1]} або {[1][2][3]} або {[3][1][3][1][2]}.

Повторення клаузи у семантичному визначенні означає повторний спуск по піддереву вказаного нетерміналу.

Призначення розділювачів:

_ – розділення вихідного тексту в межах рядка;

α – перехід на новий рядок.

Приклад фрагменту вихідного коду на мові асемблера і відповідного йому семантичного визначення:

$$\begin{array}{l} \text{MOV AX,A} \quad \leftrightarrow \quad \{\alpha \text{MOV_AX,A} \alpha \text{MOV_B,AX}\} \\ \text{MOV B,AX} \end{array}$$