**Разновидности алгоритмов разбора**

Якщо підбір кісток вдається здійснювати так, що одного разу поставлену кістку ніколи не доведеться прибирати, ми маємо справу з детермінованим алгоритмом розбору — вибір правила граматики, що застосовується, завжди однозначний. Якщо ухвалені рішення про вибір типу доміно доводиться скасовувати — алгоритм працює із поверненнями, він недетермінований. Детерміновані алгоритми ефективніші, і звичайно, завжди існує прагнення знайти та використовувати такий алгоритм для синтаксичного аналізу.

Якщо дерево будується зверху вниз від початкового нетерміналу у бік термінального ланцюжка, алгоритм відноситься до класу низхідних; від ланцюжка у бік кореня дерева - він висхідний (рис. 5).

Можна спочатку підбирати доміно для лівих символів термінального ланцюжка, а можна спочатку для правих - відповідно говорять про лівосторонні або правосторонні алгоритми розбору.



Рис.5 Початок низхідного і висхідного аналізу в граматиці G6

**Еквівалентність та однозначність граматик**

Візьмемо для прикладу граматику арифметичних виразів

G5:E→E + E|E-E|E\*E|E/E|a|b|c| (E).

Розглянемо дерева виведення термінальних ланцюжків у цій граматиці. На рис. 6 показані два різні дерева розбору виразу а + b \* с. Можливість побудувати ці дерева переконує в тому, що ланцюжок а + b \* c дійсно належить мові арифметичних виразів.



Рис.6 Дерева розбору ланцюжка а+b\*с у граматиці G5

Дерево зліва об'єднує а + b в один підвираз, який потім бере участь у ролі операнда операції множення. Таке трактування не відповідає загальноприйнятому пріоритету арифметичних операцій — множення має виконуватися раніше складання. Дерево справа представляє структуру, що відповідає правильному порядку виконання операцій. Граматика G називається однозначною, якщо будь-якої сентенції х∈ L(G) відповідає єдине дерево виведення.

Граматика G5 є неоднозначною, і це, безумовно, її серйозний недолік, який не дозволяє застосувати її на практиці для визначення мови виразів, оскільки ця граматика не дозволяє однозначним чином виявити структуру вираження. Спробуємо побудувати однозначну граматику виразів. Для цього використовуємо ще один нетермінал, позначивши його X

G7: E→X|E + X|E-X|E\*X|E/X

Х→а|b|с| (Е)



Рис.7 Дерево аналізу виразу а + b \* с в граматиці G7

Змістовно X можна розуміти як операнд виразу. Тепер для ланцюжка а+b\*с можна побудувати єдине дерево розбору. Воно показано на рис. 7 ліворуч. Можна переконатися, що у граматиці G7 єдине дерево виведення відповідає будь-якому правильному виразу. Граматика G7 однозначна.

Справа на малюнку показано редуковане (спрощене) дерево того ж виразу в граматиці G7 Такі дерева можуть служити для представлення структури висловів у трансляторах. Називатимемо їх семантичними деревами. Семантичне дерево може бути отримане з дерева розбору усуненням нетермінальних вершин і розміщенням знаків операцій у внутрішні вершини, тоді як операнди залишаються листям дерева.

Незважаючи на однозначність, G7 непридатна для використання у трансляторі, оскільки приписує виразам невідповідну структуру. Вже з прикладу ланцюжка а + b\* с (див. рис. 7) видно, що операції та операнди зв'язуються неправильно. Неважко помітити, що G7 передбачає виконання операцій без урахування їхнього пріоритету в порядку зліва направо.

Виправити становище можна, визначивши нетермінали для двох категорій виразів — доданків та множників. Після цього вираз представляється як послідовність доданків, розділених знаками плюс та мінус. У свою чергу доданки утворюються з елементарних операндів - множників, з'єднаних знаками множення та ділення.

Доданки позначимо Т (від term - елемент), множники - М (від multiplier).

G8: E→T|E+T|E-T

T→M|T\*M|T/M

М→ а |b|с| (Е)

Дерево виведення ланцюжка а + b \* с граматикою G8 показано на рис. 8. Ця граматика однозначна та приписує арифметичним виразам структуру, що відповідає правильному порядку виконання операцій.



Рис.8 Дерево вираження а+b\*с у граматиці G8

Всі три розглянуті граматики виразів (G5, G7, G8), хоч і розрізняються і мають різні властивості, задають ту саму мову.

***Граматики називаються еквівалентними, якщо породжують ту саму мову.***

Граматики G5, G7, G8 еквівалентні, оскільки L(G5) = L(G7) = L(G8).

**Ієрархія граматик Н. Хомского**

М. Хомський запропонував поділ граматик, що породжують, на чотири типи в залежності від виду їх правил.

**Тип 0**. Довільні граматики. На вигляд їхніх правил не накладається жодних обмежень. Правила мають вигляд:

**α→β**,

де **α,β** - ланцюжки терміналів та нетерміналів. Ланцюжок **α** не повинен бути порожнім.

**Тип 1**. Контекстно залежні граматики. Правила таких граматик мають вигляд:

**αAβ → αγβ**,

де **α, β, γ**-ланцюжки терміналів та нетерміналів; А – нетермінальний символ. Такий вид правил означає, що нетермінал А може бути замінений ланцюжком **γ** тільки в контексті, утвореному ланцюжками **α** та **β**.

**Тип 2**. Контекстно-вільні граматики. Їхні правила мають вигляд:

**А → γ**,

де, **А** - нетермінал; **γ** - ланцюжок терміналів та нетерміналів. Характерна риса - у лівій частині правил завжди один нетермінальний символ.

**Тип 3**. Автоматні граматики. Усі правила автоматних граматик мають одну з трьох форм:

**А→аВ**

**А→а**

**А →ε,**

де **А, В** - нетермінали; **а** - термінал; **ε** - порожній ланцюжок.

***Автоматні граматики називають також регулярними.***

Як можна бачити, граматики типу 1 є окремим випадком граматик типу 0, граматики типу 2 — окремий випадок контекстно-залежних граматик, автоматні — окремий випадок контекстно-вільних. Тобто граматика типу 3 є і граматикою типу 2 і типу 1 і типу 0.

Однак надалі, якщо не обговорено особливо, матиметься на увазі, що, наприклад, контекстно-вільною називається граматика, яка не є автоматною. Мови, що породжуються граматиками типу 0-3, називаються відповідно мовами без обмежень, контекстно-залежними, контекстно-вільними та автоматними (регулярними) мовами.

Але контекстно-вільною мовою називають мову, для якої існує його контекстно-вільна, але не автоматна граматика. Такий самий підхід застосовується до контекстно-залежних мов та мов без обмежень.

**Приклади граматик різних типів**

Розглянемо граматику G2, що породжує мову L2 = {anbncn | n > 0}.

G2: S → aSBc (тип 2)

 S → abc (тип 2)

 сВ → Bс (тип 0)

 bВ→bb (тип1)

 S → е (тип 3)

Праворуч біля кожного правила помічено тип граматики, до якої воно може бути віднесене. Типом граматики природньо вважати мінімальний із типів її правил. Отже, граматика G2 – це граматика типу 0 – довільна. Прикладом контекстно-вільної граматики може бути граматика арифметичних виразів. За допомогою контекстно-вільних граматик задається синтаксис мов програмування. Граматики цього класу докладно обговорюватимуться і надалі, зараз візьмемо конкретний приклад.

G9: N→a (1)

 N→Na (2)

 N→Nb (3)

Це граматика типу 2, оскільки правила (2) та (3) відносяться саме до цього типу. Розглянемо мову L9 = L(G9), що породжується цією граматикою. Ланцюжок належить L9 за правилом (1). Якщо до правильної пропозиції N мови приписати праворуч символ a, то знову вийде правильна пропозиція (за правилом (2)). Аналогічно, приписування до N символу b знову пропонує мову L9.

Ланцюжки, що належать мові L9, починаються символом **а**, за яким можуть слідувати **а** і **b** у довільному порядку. Якщо під **а** розуміти будь-яку латинську букву, a **b** сприймати як цифру, то G9 вважатимуться граматикою ідентифікаторів. Вона породжує послідовності літер та цифр, що починаються з літери, які використовуються у мовах програмування в ролі імен змінних, типів тощо.

Спираючись на таке змістовне трактування G9 і L9, спробуємо сконструювати автоматну граматику, що породжує мову ідентифікаторів. Перше правило граматики G9 може бути збережено. Воно, по-перше, відповідає одному з допустимих видів правил автоматних граматик, по-друге, визначає, що ідентифікатор, що складається з одного символу, може бути лише літерою.

N→a (1)

Нетермінал N це початковий символ нашої граматики. Він і позначає саме поняття «ідентифікатор». Як синонім терміна "ідентифікатор" будемо використовувати також слово "ім'я". Можна вважати, що назва N походить від Name - ім'я. Позначимо В частину ідентифікатора, яка може слідувати за першою літерою. Тоді можна записати правило (2):

N→aB (2)

Запишемо правила для В. «Хвіст» може бути буквою чи цифрою:

В→а (3)

В→b (4)

Якщо В це «хвіст» ідентифікатора, то, записавши його за буквою або цифрою, знову отримаємо правильний «хвіст»:

В →аВ (5)

В → bВ (6)

Позначимо сконструйовану граматику G10. Вона автоматна, оскільки її правила задовольняють обмеженням автоматних граматик.

 G10: N→a (1)

N→aB (2)

В→а (3)

В→b (4)

В→аВ (5)

B→bВ (6)

Граматика G10 еквівалентна G9, оскільки породжує ту саму мову:

L(G10)=L(G9).