Лекція №3 Граматики.

Формальне визначення граматики. Типи граматик і їх властивості.

Для нас найбільший інтерес представляє одна з систем генерації мов - граматики. Поняття граматики спочатку було формалізовано лінгвістами при вивченні природних мов. Передбачалося, що це може допомогти при їх автоматичної трансляції. Однак, найкращі результати в цьому напрямку досягнуто при описі не природних мов, а мов програмування. Прикладом може служити спосіб опису синтаксису мов програмування за допомогою БНФ - форми Бекуса-Наура.

Визначення. Граматика - це четвірка G = (N, T, P, S), де

N - алфавіт нетермінальних символів;

T - алфавіт термінальних символів, NT =;

P - кінцева множина правил виду, де (NT) * N (NT) *, (NT) *;

SN - початковий символ (або аксіома) граматики.

Ми будемо використовувати великі латинські літери для позначення нетермінальних символів, малі латинські букви з початку алфавіту для позначення термінальних символів, малі латинські букви з кінця алфавіту для позначення ланцюжків з T * і, нарешті, малі грецькі літери для позначення ланцюжків з (NT) *.

Будемо використовувати також скорочений запис A 1 | 2 | ... | n для позначення групи правил A 1, A 2, ..., A n.

Визначимо на множинаі (NT) * бінарне відношення виводимості таким чином: якщо P, то для всіх, (NT) *. Якщо 1 2, то говорять, що ланцюжок 2 безпосередньо виведена з 1.

Ми будемо використовувати також рефлексивно-транзитивне і транзитивне замикання відношення, а також його ступінь k 0 (позначаються відповідно *, + і k). Якщо 1 * 2 (1 +2, 1 k2), то говорять, що ланцюжок 2 виведена (нетривіально виведена, виведена за k кроків) з 1.

Якщо k (k 0), то існує послідовність кроків де = 0 і = k. Послідовність ланцюжків 0, 1, 2, ..., k в цьому випадку називають виведенням з.

Сентенціальний формою граматики G називається ланцюжок, що виводиться з її початкового символу.

Мовою, породжуваною граматикою G (позначається L (G)), називається множина всіх її термінальних сентенціальних форм, тобто

Граматики G1 і G2 називаються еквівалентними, якщо вони породжують одну й ту саму мову, тобто L (G1) = L (G2).

Приклад 2.5. Граматика G = ({S, B, C}, {a, b, c}, P, S), де

P = {S aSBC, S aBC, CB BC, aB ab, bB bb, bC bc, cC cc}, породжує мову L (G) = {anbncn | n> 0}.

Дійсно, застосовуємо n-1 раз правило 1 і отримуємо an-1S (BC) n-1, потім один раз правило 2 і отримуємо an (BC) n, потім n (n-1) / 2 раз правило 3 і отримуємо anBnCn.

Потім використовуємо правило 4 і отримуємо anbBn-1Cn. Потім застосовуємо n-1 раз правило 5 і отримуємо anbnCn. Потім застосовуємо правило 6 і n - 1 раз правило 7 і отримуємо anbncn. Можна показати, що мова L (G) складається з ланцюжків тільки такого виду.

Приклад 2.6. Розглянемо граматику G = ({S}, {0, 1}, {S 0S1, S 01}, S). Легко бачити, що ланцюжок 000111 L (G), так як існує висновок

Неважко показати, що граматика породжує мову L (G) = {0n1n | n> 0}.

Приклад 2.7. Розглянемо граматику G = ({S, A}, {0, 1}, {S 0S, S 0A, A 1A, A 1}, S). Неважко показати, що граматика породжує мову L (G) = {0n1m | n, m> 0}.

Типи граматик і їх властивості

Розглянемо класифікацію граматик (запропоновану Н. Хомського), засновану на вигляді їх правил.

Визначення. Нехай дана граматика G = (N, T, P, S). Тоді

якщо правила граматики не задовольняють жодним обмеженням, то її називають граматикою типу 0, або граматикою без обмежень.

якщо

кожне правило граматики, крім S e, має вигляд, де | | | |, і

в тому випадку, коли S e P, символ S не зустрічається в правих частинах правил,

то граматику називають граматикою типу 1, або нескороченною.

якщо кожне правило граматики має вигляд А, де АN, (NT) *, то її називають граматикою типу 2, або контекстно-вільною (КС-граматикою).

якщо кожне правило граматики має вигляд або А хB, або А х, де А, BN, х Т * то її називають граматикою типу 3, або праволінійною.

Легко бачити, що граматика в прикладі 2.5 - нескороченною, в прикладі 2.6 - контекстно-вільна, в прикладі 2.7 - праволінійна.

Мова, породжуванаграматикою типу і, називають мовою типу і. Мова типу 0 називають також мовою без обмежень, мова типу 1 - контекстно-залежним (КЗ), мова типу 2 - контекстно-вільним (КС), мова типу 3 - праволінейним.

Теорема 2.1. Кожна контекстно-вільна мова може бути породжений неукорачівающей контекстно-вільною граматикою.

Доказ. Нехай L - контекстно-вільна мова. Тоді існує контекстно-вільна граматика G = (N, T, P, S), що породжує L.

Побудуємо нову граматику G '= (N', T, P ', S') таким чином:

1. Якщо в P є правило виду A 0B11 ... Bkk, де k 0, Bi + e для 1 ik, і ні з одного ланцюжка j (0 jk) не виводиться e, то включити в P 'всі правила (крім A e) виду

де Xi - це або Bi, або e.

2. Якщо S + e, то включити в P 'правила S' S, S 'e і покласти N' = N {S '}. В іншому випадку покласти N '= N і S' = S.

Чи породжує граматика порожню ланцюжок можноо встановити наступним простим алгоритмом:

Крок 1. Будуємо множину N_0 = N | N --> e

Крок 2. Будуємо множину N_i = N | N-->$\alpha \in$ Ni - 1*

Крок 3. Якщо N_i = N_i-1, перейти до кроку 4, інакше крок 2.

Крок 4. Якщо S -->Ni значить S --> e/

Легко бачити, що G '- неукорачівающая граматика. Можна показати по індукції, що L (G ') = L (G). __

Нехай Ki - клас всіх мов типу i. Доведено, що справедливо наступне (строге) включення:

K3$\subset$ K2 $\subset$K1 $\subset$K0.

Зауважимо, що якщо мова породжується деякої граматикою, це не означає, що він не може бути породжений граматикою з більш сильними обмеженнями на правила. Наведений нижче приклад ілюструє цей факт.

Приклад 2.8. Розглянемо граматику G = ({S, A, B}, {0, 1}, {S $\rightarrow$ AB, A $\rightarrow$ 0A, A $\rightarrow$ 0, B $\rightarrow$ 1B, B $\rightarrow$ 1}, S). . Ця граматика є контекстно-вільною. Легко показати, що L(G) = $\{0^n 1^m | n, m > 0\}$.

Однак, в прикладі 2.7 наведено праволінейная граматика, що породжує ту ж мову.

Покажемо що існує алгоритм, що дозволяє для довільного КЗ-мови L в алфавіті T, і довільної ланцюжка w T * визначити, чи належить w мові L.

Теорема 2.2. Кожен контекстно-залежний мову є рекурсивним мовою.

Доказ. Нехай L - контекстно-залежний мову. Тоді існує деяка неукорачівающая граматика G = (N, T, P, S), що породжує L.

Нехай w T * і | w | = n. Якщо n = 0, тобто w = e, то приналежність w L перевіряється тривіальним чином. Так що будемо припускати, що n> 0.

Визначимо множина Tm як множина рядків $u \in (N \cup T)^+$довжини не більше n таких, що висновок S * u має не більше m кроків. Ясно, що T0 = {S}.

Легко показати, що Tm можна отримати з Tm-1 переглядаючи, які рядки з довжиною, меншою або рівною n можна вивести з рядків з Tm-1 застосуванням одного правила, тобто

$$T_m = T_{m-1} \cup \{u \mid v \Rightarrow u \text{ для некоторого } v \in T_{m-1}, \text{ где } |u| \leqslant n\}.$$

Якщо $S \Rightarrow^* u$ и $|u| \leqslant n$, то $u \in T_m$ для деякого m. Якщо з S не виводиться u або | u |> n, то u не належить Tm ні для якого m.

Очевидно, що $T_m \supseteq T_{m-1}$ для всіх m 1. Оскільки Tm залежить тільки від $m \geqslant 1$, якщо Tm = Tm-1, то Tm = Tm +1 = Tm +2 = ... Процедура буде обчислювати T1, T2, T3,. . . поки для деякого m не виявиться Tm = Tm-1. Якщо w не належить Tm, то не належить і L (G), оскільки для j> m виконано Tj = Tm. Якщо $w \in T_m$, то $S \Rightarrow^* w$.

Покажемо, що існує таке m, що Tm = Tm-1. Оскільки для кожного $i \geqslant 1$ справедливо $T_i \supseteq T_{i-1}$, то якщо $T_i \neq T_{i-1}$, то число елементів в Ti принаймні на 1 більше, ніж в Ti-1. Нехай $|N \cup T| = k$. Тоді число рядків в $(N \cup T)^+$довжини меншою або рівною n одно $k + k^2 + ... + k^n \leqslant nk^n$. Тільки ці рядки можуть бути в будь-якому Ti. Значить, Tm = Tm-1 для деякого m nkn. Таким чином, процедура, що обчислює Ti для всіх i 1 до тих пір, поки не будуть знайдені два рівних множинаі, гарантовано закінчується, значить, це алгоритм.