3.7. Контекстно-свободные грамматики. Нормальная форма Хомского. Общие методы разбора.

Определение

КС-грамматика G – это четверка: $G = \{V, T, P, S\}$

V – множество переменных (нетерминалов),

T – множество токенов (терминалов),

P – множество продукций вида $A \to \alpha_1\alpha_2\ldots\alpha_k$, где A – нетерминал, каждый из α_i – терминал либо нетерминал. Правая часть продукции может быть пустой (ε).

S – стартовый символ.

Пример 1: язык палиндромов $G_{pal} = \{\{P\}, \{0,1\}, A, P\}$, где A – множество продукций:

$P \to \varepsilon$
$P \to \mathbf{0}$
$P \to \mathbf{1}$
$P \to \mathbf{0}P\mathbf{0}$
$P \to \mathbf{1}P\mathbf{1}$

Пример 2:

$E \to I$,
$E \to E + E$, $E \to E * E$, $E \to (E)$,
$I \to \boldsymbol{a}$, $I \to \boldsymbol{b}$, $I \to I\boldsymbol{a}$, $I \to I\boldsymbol{b}$, $I \to I\mathbf{0}$, $I \to I\mathbf{1}$.

КС-язык – язык, порождаемый КС-грамматикой.
NB любой регулярный язык (т.е. задаваемый регулярным выражением) является КС.

Общие методы разбора:
1. Приведение к нормальной форме Хомского.
2. Алгоритм Кока-Янгера-Хасами.

Нормальная форма Хомского

Рассмотрим контекстно-свободную грамматику Γ, из которой удалены бесполезные символы, ε-правила, длинные правила и цепные правила. Такая грамматика содержит только правила следующего вида:

- $A \to BC$
- $A \to Bc$
- $A \to bC$
- $A \to bc$
- $A \to a$
- возможно, $S \to \varepsilon$ (при условии, что S не содержится в правых частях правил)

Избавимся от правил, в правых частях которых записаны два символа, один из которых является терминалом, то есть правил вида $A \to Bc$, $A \to bC$ и $A \to bc$.

Введем для каждого терминала a "персональный" нетерминал N_a. Затем заменим:

- $A \to Bc$ => $A \to BN_c$; $N_c \to c$;
- $A \to bC$ => $A \to N_bC$; $N_b \to b$;
- $A \to bc$ => $A \to N_bN_c$; $N_b \to b$; $N_c \to c$.

3.7

Теперь у нас остались только правила вида $A \to BC$, $A \to a$ и, возможно, $S \to \varepsilon$ (при условии, что S не содержится в правых частях правил). Грамматика, содержащая правила только такого вида, называется грамматикой в **нормальной форме Хомского**.

Заметим, что любую контекстно-свободную грамматику можно привести к нормальной форме Хомского. Такая форма грамматики очень удобна для работы многих алгоритмов над грамматиками, например, алгоритм Кока-Янгера-Касами

Алгоритм Кока-Янгера-Касами разбора грамматики в НФХ

Пусть дана контекстно-свободная грамматика Γ и слово $w \in \Sigma^*$. Требуется выяснить, выводится ли это слово в данной грамматике.

Алгоритм для НФХ-грамматики

Пусть Γ приведена к нормальной форме Хомского.

Пусть $a_{A,i,j} = true$, если из нетерминала A можно вывести подстроку $w[i..j]$. Иначе $a_{A,i,j} = false$:

$$a_{A,i,j} = \begin{cases} true, & A \Rightarrow^* w[i..j]; \\ false, & \text{else}. \end{cases}$$

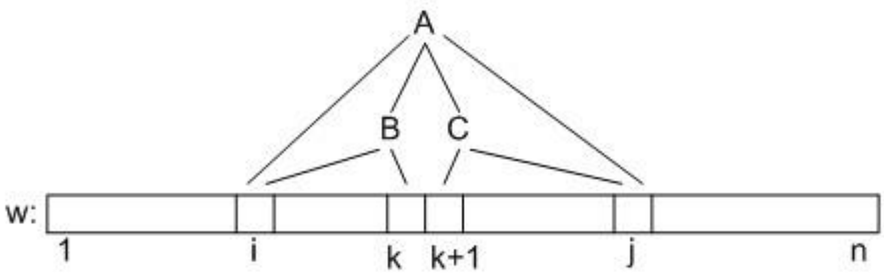
Будем динамически заполнять матрицу $a_{A,i,j}$ следующим алгоритмом:

- **База**. Ячейки $a_{A,i,i}$ заполняются истиной, если правило $A \to w[i]$ принадлежит множеству правил P грамматики Γ:

$$a_{A,i,j} = [A \to w[i] \in P].$$

- **Переход**. Пусть на текущем шаге $j - i = m > 0$. Если все ячейки, для которых справедливо $j - i < m$, уже вычислены, то алгоритм смотрит, можно ли вывести подстроку $w[i..j]$ из этих ячеек:

$$a_{A,i,j} = \bigvee_{k=i}^{j-1} \bigvee_{A \to BC} (a_{B,i,k} \wedge a_{C,k+1,j}).$$



- **Завершение**. После окончания работы ответ содержится в ячейке $a_{S,1,n}$, где $n = |w|$.

3.7

Сложность алгоритма

Необходимо вычислить n^2 булевых величин. На каждую требуется затратить $n \cdot |P_A|$ операций, где $|P_A|$ – количество правил. Суммируя по всем правилам получаем конечную сложность $O\left(n^3 \cdot |\Gamma|\right)$.

Алгоритму требуется $n^2 \cdot |N|$ памяти, где $|N|$ – количество нетерминалов грамматики.

Минус алгоритма заключается в том, что изначально грамматику необходимо привести к НФХ.