

---

## Контекстно-вільні граматики. Комп'ютерне завдання

---

Святослав Хмелевський, ФІ-82

22 грудня 2021 р.

Завдання: На вхід алгоритму подаються еталонні зображення  $X_0 : P \rightarrow \{0, 1\}$ ,  $X_1 : P \rightarrow \{0, 1\}$  цифр 0 та 1 відповідно, де  $P = \{1, \dots, h\} \times \{1, \dots, w\}$  – множина пікселів (зображення мають однаковий розмір). Користувач подає на вхід бінарне зображення  $X : P' \rightarrow \{0, 1\}$ , де  $P' = \{1, \dots, 3 \cdot h\} \times \{1, \dots, c \cdot w\}$  – множина пікселів зображення, де  $c \in \mathbb{N}$  – кількість стовпчиків з цифрами у зображенні. Програма має відповідати, чи містить подане на вхід зображення коректне додавання у стовпчик. Якщо містить, то потрібно сказати, чи є у найлівішому стовпчику перенесення одиниці. Перевірка коректності має відбуватися за допомогою двовимірних контекстно-вільних граматик. Алгоритм повинен мати поліноміальну залежність часу роботи від кількості стовпців зображення.

Розв'язок: Побудова відповідної контекстно-вільної граматики.

$$\begin{aligned} I &\rightarrow V_0 I, \quad I \rightarrow V_1 I', \quad I' \rightarrow V'_0 I, \quad I' \rightarrow V'_1 I', \\ I &\rightarrow V_0, \quad I' \rightarrow V'_0, \quad Z \rightarrow 0, \quad F \rightarrow 1, \\ V_0 &\rightarrow \frac{A_1}{Z}, \quad V_0 \rightarrow \frac{A_2}{F}, \quad V'_0 \rightarrow \frac{A_3}{Z}, \\ V_1 &\rightarrow \frac{A_1}{F}, \quad V'_1 \rightarrow \frac{A_2}{Z}, \quad V'_1 \rightarrow \frac{A_3}{F}, \\ A_1 &\rightarrow \frac{Z}{Z}, \quad A_2 \rightarrow \frac{Z}{F}, \quad A_2 \rightarrow \frac{F}{Z}, \quad A_3 \rightarrow \frac{F}{F}. \end{aligned}$$

Ці правила можна представити в наступному вигляді.

$$T = \{0, 1\}, \quad K = \{I, I', V_0, V'_0, V_1, V'_1, A_1, A_2, A_3, Z, F\},$$

$$P_r(Z, 0) = P_r(F, 1) = P_r(I, V_0) = P_r(I', V'_0) = 1,$$

в інших випадках  $P_r$  дорівнює 0,

$$P_h(V_0, I, I) = P_h(V_1, I', I) = P_h(V'_0, I, I') = P_h(V'_1, I', I') = 1,$$

в інших випадках  $P_h$  дорівнює 0,

$$\begin{aligned} P_v(A_1, Z, V_0) &= P_v(A_2, F, V_0) = P_v(A_3, Z, V'_0) = P_v(A_1, F, V_1) = \\ &= P_v(A_2, Z, V'_1) = P_v(A_3, F, V'_1) = P_v(Z, Z, A_1) = P_v(Z, F, A_2) = \\ &= P_v(F, Z, A_2) = P_v(F, F, A_3) = 1, \text{ в інших випадках } P_v \text{ дорівнює } 0. \end{aligned}$$

Неформальний опис алгоритму на основі СУК.

## 1. Ініціалізація:

1.1 На вхід подається "зображення"  $x \in T^{3 \times c}$ .1.2  $\forall i = \overline{1, 3}, \forall j = \overline{1, c}, \forall m \in Z, F :$   
 $f(i, i, j, j, m) = P_r(m, x(i, j)).$   
(Всі інші значення нульові)

## 2. Основний цикл:

2.1 Послідовно аналізуються всі прямокутники

 $R \in \{ \langle i_1, i_2, j_1, j_2 \rangle : i_1 \leq i_2, j_1 \leq j_2, i_1, i_2 \in \{1, 2, 3\},$   
 $j_1, j_2 \in \{1, \dots, c\} \}$  (в певному ієрархічному порядку)  
і кожен символ  $k \in K$ .2.2  $f(i_1, i_2, j_1, j_2, k) \leftarrow \bigvee_{j: j_1 \leq j < j_2} \bigvee_{k_l \in K} \bigvee_{k_r \in K} (f(i_1, i_2, j_1, j, k_l) \wedge$   
 $\wedge P_h(k_l, k_r, k) \wedge f(i_1, i_2, j + 1, j_2, k_r)).$ 2.3  $f(i_1, i_2, j_1, j_2, k) \leftarrow f(i_1, i_2, j_1, j_2, k) \vee \bigvee_{i: i_1 \leq i < i_2} \bigvee_{k_u \in K} \bigvee_{k_d \in K} (f(i_1, i, j_1,$   
 $j_2, k_u) \wedge P_v(k_u, k_d, k) \wedge f(i + 1, i_2, j_1, j_2, k_d)).$ 2.4  $f(i_1, i_2, j_1, j_2, k) \leftarrow f(i_1, i_2, j_1, j_2, k) \vee \bigvee_{k' \in K} (f(i_1, i_2, j_1, j_2, k') \wedge P_r(k, k')).$ 

## 3. Результат:

3.1 Якщо  $f(1, 3, 1, c, I) = 1$ , то:  
Додавання коректне без перенесення.3.2 Якщо  $f(1, 3, 1, c, I') = 1$ , то:  
Додавання коректне з перенесенням.3.3 Якщо  $f(1, 3, 1, c, I) \vee f(1, 3, 1, c, I') = 0$ , то:  
Додавання не є коректним.