Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación

IIC2223 – Teoría de Autómatas y Lenguajes Formales – 2' 2024
IIC2224 – Autómatas y Compiladores

Tarea 5

Pregunta 1

Para cada uno de los siguientes lenguajes, determine si es libre de contexto o no. Demuestre su respuesta.

1. $L_1 = \left\{a^i b^j \mid 0 < i < j < 2i\right\}$
2. $L_2 = \left\{a^i b^j c^k \mid 0 < i \cdot j = k\right\}$

Solución

Problema 1.a El lenguaje **es libre de contexto**. Para esto, se debe dar una gramática libre de contexto. Un ejemplo es la siguiente gramática G:

$$S \to a\ X\ bb$$

$$X \to a\ X\ bb \mid Y$$

$$Y \to a\ Y\ b \mid ab$$

De la definición de G uno puede ver que las derivaciones de G serán de la forma:

$$S \Rightarrow aXbb \overset{*}{\Rightarrow} a^k Y b^{2k} \overset{*}{\Rightarrow} a^k a^{l-1} Y b^{l-1} b^{2k} \Rightarrow a^{k+l} b^{2k+l}.$$

Con $k > 0$ y $l > 0$. Es fácil ver que:

$$0 < k + l < 2k + l < 2k + 2l$$

Si definimos $i = k + l$ y $j = 2k + l$ tenemos que $0 < i < j < 2i$. Por lo tanto, $\mathcal{L}(G) \subseteq L_1$. Para demostrar el otro sentido, consideremos una palabra cualquiera $a^i b^j \in L_1$ tal que $0 < i < j < 2i$. Defina $k = j - i$ y $l = 2i - j$. Entonces:

$$k + l = i$$

$$2k + l = 2j - 2i + 2i - j = j$$

Por lo tanto, $a^i b^j = a^{k+l} b^{2k+l} \in \mathcal{L}(G)$.

Distribución de puntaje

- 1 punto por decir que el lenguaje es libre de contexto
- 1 punto por plantear la gramática
- 1 punto por demostrar un sentido
- 1 punto por demostrar el otro sentido

Problema 1.b El lenguaje **no es libre de contexto**. Para demostrar esto, usamos el lema del bombeo de lenguajes libres de contexto. Sea $N > 0$. Definimos:

$$z = a^{N^2} b^N c^{N^3}.$$

Considere una partición cualquiera $uvwxy = z$ tal que: $vx \neq \varepsilon$ y $|vwx| \leq N$. Como $|vwx| \leq N$, se tiene que (1) $vwx \in \mathcal{L}(a^* b^*)$ o (2) $vwx \in \mathcal{L}(b^* c^*)$. A continuación analizaremos ambos casos:

Caso (1): Sin pérdida de generalidad, suponemos que $v \in \mathcal{L}(a^*)$ y $x \in \mathcal{L}(b^*)$ (de no ser así, al bombear la palabra estará claramente fuera del lenguaje). Entonces, se tiene:

$$\underbrace{a^k}_{u}\ \underbrace{a^l}_{v}\ \underbrace{a^m b^{m'}}_{w}\ \underbrace{b^n}_{x}\ \underbrace{b^o c^p}_{z}$$

Para algún $k, l, m, m', n, o, p \in \mathbb{N}$ tal que $k + l + m = N^2$, $m' + n + o = N$, $p = N^3$ y $l > 0$ o $n > 0$. Si escogemos $i = 2$, tenemos:

$$u\, v^2\ w\, x^2\ z = a^{k+2l+m} b^{m'+2n+o} c^p$$

Así, se tiene:

$$\begin{aligned}(k + 2l + m) \cdot (m' + 2n + o) &= (k + l + m + l) \cdot (m' + n + o + n)\\ &= (N^2 + l) \cdot (N + n)\\ &= N^3 + N \cdot l + N^2 \cdot n + l \cdot\end{aligned}$$

Como $l \neq 0$ o $n \neq 0$, entonces $N^3 + N \cdot l + N^2 \cdot n + l \cdot n > N^3$. Por lo tanto, $uv^2wx^2z \notin L_2$.

Caso (2): De nuevo, sin pérdida de generalidad, suponemos que $v \in \mathcal{L}(b^*)$ y $x \in \mathcal{L}(c^*)$. Entonces, se tiene:

$$\underbrace{a^k b^l}_{u}\ \underbrace{b^m}_{v}\ \underbrace{b^n c^{n'}}_{w}\ \underbrace{c^o}_{x}\ \underbrace{c^p}_{y}$$

Para algún $k, l, m, m', n, o, p \in \mathbb{N}$ tal que $k = N^2$, $l + m + n = N$, $n' + o + p = N^3$ y $m > 0$ o $o > 0$. Notar que $|vwx| = m + n + n' + o \leq N$. Por lo tanto, $o \leq N$. Si escogemos $i = 2$, tenemos que $uv^2wx^2y = a^k b^{l+2m+n} c^{n'+2o+p}$. Si multiplicamos el número de letras a por el número de letras b, tenemos que:

$$k \cdot (l + m + n + m) = N^2 \cdot (N + m) = N^3 + N^2 \cdot m$$

Ahora, el número de letras c es $n' + o + p + o = N^3 + o$, por lo que necesitamos demostrar que $N^2 \cdot m \neq o$. Para esto nos ponemos en dos casos: si $m \neq 0$ y si $m = 0$. Si $m \neq 0$, entonces $N^2 \cdot m > N \geq o$ y se cumple que $N^2 \cdot m \neq o$. En otro caso, si $m = 0$, se tiene $o > 0$ (por condición del lema de bombeo) y entonces $N^2 \cdot m = 0 < o$, esto es, $N^2 \cdot m \neq o$. En ambos casos, se tiene que:

$$k \cdot (l + 2m + n) \neq n' + 2o + p$$

Por lo tanto, se tiene que $uv^2wx^2y \notin L_2$.

Nota: Es importante destacar que si se elige $a^N b^N c^{N^2}$, el bombeo no funciona cuando $vwx \in \mathcal{L}(b^* c^*)$.

Distribución de puntaje

- 1 punto por decir que el lenguaje no es libre de contexto
- 1 punto por definir z de manera correcta
- 1 punto por analizar el primer caso
- 1 punto por analizar el segundo caso

Pregunta 2

Sea G una gramática en forma normal de Chomsky (CNF). Para un árbol de derivación T de G, se define $\text{depth}(T)$ como el camino más largo de la raíz hasta una hoja. Inductivamente, si $X(a)$ es un árbol con una variable X, entonces $\text{depth}(X(a)) = 1$ y si $X(t_1, t_2)$ es un árbol con X la variable en la raíz y t_1, t_2 sus subárboles, entonces $\text{depth}(X(t_1, t_2)) = \max\{\text{depth}(t_1), \text{depth}(t_2)\} + 1$.

Problema:	HIGHEST-DERIVATION
Input:	Una gramática $G = (V, \Sigma, P, S)$ en CNF y $w = a_1 \dots a_n \in \Sigma^*$
Output:	$\max\{\text{depth}(T) \mid T \text{ es un árbol de derivación de } G \text{ sobre } w\}$

Esto es, el problema HIGHEST-DERIVATION consiste en, dada una gramática G en forma normal de Chomsky y una palabra $w = a_1 \dots a_n$, calcular la mayor altura entre los árboles de derivación de G sobre w. En caso de que no existan árboles de derivación de G sobre w (es decir, $w \notin \mathcal{L}(G)$), el resultado debe ser 0.

Escriba un algoritmo que resuelva HIGHEST-DERIVATION en tiempo $\mathcal{O}(|G| \cdot |w|^3)$, donde $|G|$ es el número de variables y producciones en G. Demuestre la correctitud de su algoritmo.

Solución

Sea $G = (V, \Sigma, P, S)$ una CFG en CNF y $w = a_1 \dots a_n$. El algoritmo es una extensión del algoritmo CKY donde los conjuntos C_{ij} ahora son una función **parcial**:

$$C_{ij} : V \to \mathbb{N}$$

tal que, si $X \in \text{dom}(C_{ij})$, entonces $C_{ij}(X) = h$ significa que hay un árbol de derivación de con raíz X y altura máxima h. Por otro lado, si $X \notin \text{dom}(C_{ij})$, entonces no hay ningún árbol de derivación con raíz X.

Notemos que cada conjunto C_{ij} se puede implementar como una tabla de hash que las llaves son V y cada acceso toma tiempo constante.

Ahora, al igual que en el algoritmo CKY, completamos las entradas $C_{i,i+k}$ iterando desde $k = 0, ..., n$ y $i = 1, ..., n - k$. Lo hacemos de la siguiente manera:

- **Caso base**: Para $k = 0$, se define $C_{ij}(X) = 1$ si, y sólo si, existe $X \to a_i \in P$.
- **Caso inductivo**: Para k cualquiera, suponga que $C_{j,j+l}$ ya esta computado con $j < i$ y $l < k$.
  Entonces para cada X, se calcula:

  $$C_{i,i+k}(X) = \max_{X \to YZ \in P} \max_{i \leq j < i+k} \Big(\max\{C_{i,j}(Y), C_{j+1,i+k}(Z)\} + 1 \Big)$$

  donde si $C_{i,j}(Y)$ o $C_{j+1,i+k}(Z)$ no está definido, entonces no se calcula el último máximo (esto es, podemos suponer que su valor es $-\infty$). También si para ningún $X \to YZ \in P$ y ningún $j < k$ está definido, entonces $C_{i,i+k}(X)$ **no** está definido.

La idea es la misma que CKY pero ahora, aparte de guardar la variable X en $C_{i,i+k}$, también guardamos la altura máxima como el máximo entre el hijo izquierdo e hijo derecho más 1.

- **Tiempo**: El tiempo es $\mathcal{O}(|w|^3|G|)$, dado que en el caso inductivo para por cada variable se tiene lo siguiente:
  - $k = 0, ..., |w|$
  - $i = 1, ..., |w| - k$
  - cada $X \to YZ \in P$ (a lo más $|P|$)
  - cada $i \leq j < i + k$

  Calculamos el máximo y obtenemos que el total es $\mathcal{O}(|w|^3|G|)$.
- **Correctitud**: Lo demostraremos con inducción sobre k e i, donde la hipótesis dice:
  "$C_{i,i+k}(X) = h$ si, y solo si, existe un árbol de derivación sobre $a_i...a_{i+k}$ con raíz X y el de altura máxima entre ellos es h."
  - **Caso Base**:
    Es cierto para $k = 0$ e $i = 1, ..., n$, ya que el único árbol sobre a_i con raíz X tiene altura 1 si, y sólo si existe una producción $X \to a_i$.
  - **Caso Inductivo**:
    Suponemos que es cierto para $k' < k$ y demostramos para k. Sea T el árbol más alto con raíz X sobre $a_i...a_{i+k}$. Suponga que $T = X(t_1, t_2)$ y $raiz(t_1) = Y$ y $raiz(t_1) = Z$, y t_1 es sobre $a_i...a_j$ y t_2 es sobre $a_{j+1}...a_{i+k}$. Entonces, $X \to YZ \in P$ y se cumple que t_1 es el árbol más alto para Y o t_2 es el árbol más alto para Z (si no, T no sería el más alto). Por lo tanto, por la definición de $C_{i,i+k}$ y la definición de altura, la hipótesis en k se cumple.

Distribución de puntaje

- 1 punto por definir la función parcial C_{ij}.
- 1 punto por el caso base de la definición de C_{ij}.
- 1 punto por el caso inductivo de la definición de C_{ij}.
- 1 punto por demostrar correctitud y tiempo

Evaluación y puntajes de la tarea

Cada item de cada pregunta se evaluará con un puntaje de 0, 1, 2, 3 o 4 puntos. Todas las preguntas tienen la misma ponderación en la nota final y cada item tiene la misma ponderación en cada pregunta.