Pontificia Universidad Católica de Chile
Departamento de Ciencia de la Computación
IIC2223 – Teoría de Autómatas y Lenguajes Formales
Segundo semestre de 2024
Profesor: Cristian Riveros
Ayudante: Amaranta Salas

Ayudantia 10

Gramáticas Libre de Contexto (CFG)

Problema 1

Para cada uno de los siguientes lenguajes, muestre una gramática libre de contexto que lo defina y explique su correctitud. No es necesario demostrar su correctitud, pero si explicar de manera precisa porque la gramática propuesta cumple con lo solicitado.

1. Todas las palabras de paréntesis $\{(,)\}$ bien balanceadas, tal que el número de paréntesis abiertos es par. Por ejemplo, $(())$ y $(()())()$ están en el lenguaje, pero $(()))$ y $()(())$ no lo están.
2. Todas las palabras de letras en $\{a, b\}$ de largo par, tal que el número de ocurrencias de b-letras en posiciones pares es igual al número de b-letras en posiciones impares. Por ejemplo, $baab$ y $ababbaba$ están en el lenguaje, pero bba y $baba$ no lo están.

Solución

1. Se define la gramática como:

$$\begin{aligned} \mathcal{G} : S_0 &\to (S_1) \mid S_0S_0 \mid S_1S_1 \mid \epsilon \\ S_1 &\to (S_0) \mid S_1S_0 \mid S_0S_1 \end{aligned}$$

   Respecto a la explicación de correctitud, explicaremos primero la generación de paréntesis bien balanceados y luego la generación de paréntesis abiertos pares.

   Por un lado, para generar palabras con paréntesis bien balanceados, necesitamos producciones que tengan el formato $S \to (S)$ y $S \to SS$, las cuales permitirán generar todas las palabras posibles que estén bien balanceadas.

   Sin embargo, para asegurar que la cantidad de paréntesis abiertos sea par, necesitamos trabajar con dos variables, S_0 y S_1. Las producciones de S_0 generan palabras con un número par de paréntesis, mientras que las de S_1 generan palabras con un número impar de paréntesis. Por esto mismo, solo es posible llegar a ϵ desde S_0 para asegurar la correctitud de la gramática.

2. Se define la gramática como:

$$\mathcal{G} : \ S \to bb \mid aa \mid SS \mid baSab \mid abSba \mid \epsilon$$

   Respecto a la explicación de correctitud, este ejercicio puede verse como una variación de una gramática $\mathcal{G}_0$ que permite igualar la cantidad de letras a y letras b, cuyas derivaciones se muestran a continuación:

$$\mathcal{G}_0 : \ S \to aSb \mid bSa \mid SS \mid \epsilon$$

   Para adaptarlo en este caso, podemos seccionar cualquier palabra del lenguaje en conjuntos de pares de letras, al ser las palabras de largo par. De esta forma, es posible analizar en cada sección la existencia de una letra b en la posición par y/o en la posición impar. Por ejemplo, podemos seccionar la siguiente palabra:

$$bbbabbbaaaabababba$$

de la forma:

$$bb|ba|bb|ba|aa|ab|ab|ab|ba$$

Y en cada sección revisar la cantidad de b en la posición par e impar. Al sumar estos conteos a lo largo de la palabra, deberíamos llegar al mismo valor en posiciones pares e impares para que esta pertenezca al lenguaje.

Por esto, las derivaciones creadas añaden siempre dos letras, y en el caso de las derivaciones que incluyen variables, dos letras tanto a la izquierda como a la derecha, de forma que si en un extremo hay una b en posición impar, en el otro extremo habrá una en posición par, y viceversa. Así es posible asegurar que siempre la cantidad de b pares será igual a la de impares.

Problema 2

Sea $\Sigma = \{a, b\}$. Una palabra $w \in \Sigma^*$ es un palíndromo si $w = w^r$, donde w^r es el reverso de w. Decimos que w es un casi-palíndromo si al eliminar o cambiar exactamente una letra en cualquier posición de w, entonces el resultado es un palíndromo. Por ejemplo, la palabra $aabba$ es un casi palíndromo debido a que podemos eliminar el segundo símbolo se obtiene $abba$ que sí es un palíndromo. Sea L el lenguaje de todos los casi-palíndromos. Construya una gramática libre de contexto $\mathcal{G}$ tal que $\mathcal{L}(\mathcal{G}) = L$ y demuestre que su gramática $\mathcal{G}$ es correcta, esto es, demuestre que $\mathcal{L}(\mathcal{G}) = L$.

Solución

Una posible gramática $\mathcal{G}$ que cumple con el enunciado es:

$$\begin{aligned} \mathcal{G}: \quad S &\to aSa \mid bSb \mid X \mid X^{'} \\ X &\to aY \mid bY \mid Ya \mid Yb \\ X^{'} &\to aYb \mid bYa \\ Y &\to aYa \mid bYb \mid a \mid b \mid \varepsilon \end{aligned}$$

Como complemento de la construcción, uno podría explicar que la construcción de la gramática es la siguiente: Desde un estado inicial S se produce un palíndromo hasta que ocurre uno de los dos posibles fallos. El primero (variable X) es un casi-palíndromo porque "falla" con un símbolo extra a la izquierda o derecha, que puede ser 'a' o 'b', el cual debe ser eliminado para que la palabra sea un palíndromo. El segundo fallo (variable $X^{'}$) es una asimetría entre los dos símbolos en una posición simétrica. Por último, la variable Y produce siempre palíndromos, y así, solo ocurrirá 1 solo fallo para que sea casi-palíndromo.
Primero vemos que de la gramática anterior se deduce que:

$(*)$ $S \to^* uSv$ si, y solo si, $|u| = |v|$ y uv es un palíndromo.
$(**)$ $Y \to^* w$ si, y solo si, w es un palíndromo.

Estas dos afirmaciones las usaremos en la siguiente demostración, cabe recalcar que no son necesarias, solo facilitan la demostración más adelante. Además, estas afirmaciones son dependientes de la construcción realizada en la primera parte.

PD: $L \subseteq \mathcal{L}(G)$
Sea $w \in L$. Debemos demostrar que existe una derivación en $\mathcal{G}$ que produzca w (recordemos en este paso que w es una palabra arbitraria, por lo que estamos demostrando para todo w). Tenemos dos casos posibles para una palabra casi-palíndromo:

1. $w = ucv$ con $c \in \{a, b\}$ y uv es un palíndromo, con fallo del primer tipo con u no necesariamente igual a v

2. $w = uc_1zc_2v$ con $c_1 \neq c_2, |u| = |v|$ y uc_1zc_1v es palíndromo con fallo del segundo tipo

Para el caso 1, si $w = ucv$, sin pérdida de generalidad asumimos que $|u| \leq |v|$. Descomponemos $v = v_1v_2$ tal que $|u| = |v_2|$, con uv_2 palíndromo (notemos que dado que uv es palíndromo siempre podemos hacer esta descomposición y notar que elegimos la segunda parte de v). De esto, se desprende que v_1 es un palíndromo. Finalmente, usando (*) y (**) existe una derivación:

$$S \rightarrow^* uSv_2 \rightarrow uXv_2 \rightarrow ucYv_2 \rightarrow^* ucv_1v_2$$

Para el caso 2, si $w = uc_1zc_2v$. Sabemos que uv es palíndromo y z es palíndromo. Sin pérdida de generalidad, asumimos que $c_1 = a$ y $c_2 = b$ (esto se puede realizar porque en el resto de casos elegiríamos otra producción que ya existe). Finalmente, usando (*) y (**) existe una derivación:

$$S \rightarrow^* uSv \rightarrow uX^{'}v \rightarrow uaY^{'}bv \rightarrow^* uazbv$$

PD: $\mathcal{L}(G) \subseteq L$
Sea $w \in \mathcal{L}(G)$. Por construcción tenemos dos posibilidades (2 fallos):

1. $S \rightarrow^* uSv \rightarrow uXv \rightarrow ucYv \rightarrow^* uczv$, con uzv palíndromo, donde el caso en que c está a la derecha es similar. Luego $w \in L$ ya que eliminando c obtenemos un palíndromo.

2. Sin pérdida de generalidad: $S \rightarrow^* uSv \rightarrow uX^{'}v \rightarrow uaYbv \rightarrow^* uazbv$, con $uazav$ palíndromo, en cualquiera de las dos posibilidades de asimetría, con un cambio de símbolo obtenemos un palíndromo. Luego $w \in L$, debido a que mediante una sola sustitución podemos obtener un palíndromo.

Problema 3

Sea Σ un alfabeto. Considere $u = a_1 \dots a_n$ y $v = b_1 \dots b_m$ dos palabras sobre Σ tal que $u \neq v$ y $1 \leq n \leq m$. Considere el autómata finito no-determinista $\mathcal{A}_{u,v} = (Q, \Sigma, \Delta, I, F)$ tal que:

- $Q = (\{u\} \times \{1, \dots, n\}) \cup (\{v\} \times \{1, \dots, m\}) \cup \{0\}$
- $\Delta = \{(0, a, 0) \mid a \in \Sigma\} \cup$
  $\{(0, a_1, (u, 1)), (0, b_1, (v, 1))\} \cup$
  $\{((u, i), a_{i+1}, (u, i + 1)) \mid 1 \leq i < n\} \cup$
  $\{((v, i), b_{i+1}, (v, i + 1)) \mid 1 \leq i < m\}$
- $I = \{0\}$
- $F = \{(u, n), (v, m)\}$

Sea $\mathcal{A}_{u,v}^{det} = (Q^{det}, \Sigma, \delta^{det}, \{0\}, F^{det})$ la determinización de $\mathcal{A}_{u,v}$ tal que Q^{det} contiene solo los estados alcanzables desde $\{0\}$. Demuestre que $|Q^{det}| \leq n + m + 1$.

Solución

Para resolver el problema vamos a extender la demostración del teorema de determinizar un autómata de un patrón vista en clases (clase 18). Sea $c_1 \dots c_M$ la representación de cualquiera de la palabras $u = a_1 \dots a_n$ o $v = b_1 \dots b_m$ que se utilizará para cubrir todos los casos en la demostración. Luego, la propiedad a demostrar es:

Para todo $S \in Q^{det}$, que son todos los estados alcanzables, y $(c_1 \dots c_k, M) \in S$ tal que $M = \max_{(\cdot,i) \in S} i$ con $w = c_1 \dots c_k \in \{u, v\}$, se cumple que:

$$(d_1 \dots d_{k'}, i) \in S \text{ si, y solo si, } d_1 \dots d_i \text{ es sufijo de } c_1 \dots c_M$$

PD: $(d_1 \dots d_{k'}, i) \in S \Rightarrow d_1 \dots d_i$ es sufijo de $c_1 \dots c_M$
Sea $(d_1 ... d_{k'}, i) \in S$. Como S es alcanzable, por la construcción de la determinización, existe un camino de $\{0\}$ a S leyendo w en $\mathcal{A}^{det}$. Como $(d_1 ... d_{k'}, i) \in S$ entonces existe una ejecución en $\mathcal{A}$ de la forma

$$0 \to 0 \to \cdots \to 0 \xrightarrow{d_1} (\cdot, 1) \xrightarrow{d2} \dots \xrightarrow{d_i} (d_1 \dots d_{k'}, i)$$

Y como $(c_1 \dots c_k, M) \in S$ entonces existe una ejecución en $\mathcal{A}$ de la forma

$$0 \to 0 \to \dots \to 0 \xrightarrow{c_1} (\cdot, 1) \xrightarrow{c_2} \dots \xrightarrow{c_k} (c_1 \dots c_k, M)$$

En ambos casos se observa que se las ejecuciones repiten muchas veces el estado 0 hasta salir de él leyendo la palabra. Se observa que $c_1 \dots c_M$ es sufijo de w y $d_1 \dots d_i$ es sufijo de w. Como $i \leq M$, entonces $d_1 \dots d_i$ es también sufijo de $c_1 \dots c_M$.

PD: $d_1 \dots d_i$ es sufijo de $c_1 \dots c_M \Rightarrow (d_1 \dots d_{k'}, i) \in S$
Se tiene que $d_1 \dots d_i$ es sufijo de $c_1 \dots c_M$, además sabemos que $(c_1 \dots c_k, M) \in S$. De forma similar a lo anterior, sabemos que existe una ejecución en $\mathcal{A}^{det}$ tal que de $\{0\}$ se llega a S leyendo w. Y en el autómata original $\mathcal{A}$, se tiene una ejecución:

$$0 \to 0 \to \dots \xrightarrow{c_1 \dots c_M} (c_1 \dots c_M, M)$$

Como $d_1 \dots d_i$ es sufijo de $c_1 \dots c_M$, y $c_1 \dots c_M$ es sufijo de w, por transitividad, se tiene que $d_1 \dots d_i$ es sufijo de w.
De este modo se puede construir la ejecución $0 \to 0 \to \dots \to 0 \xrightarrow{d_1 \dots d_i} (d_1 \dots d_{k'}, i)$ en $\mathcal{A}$. Donde la palabra leıda es w, por lo tanto hemos construido una ejecución que llega al estado deseado, por lo tanto

$$(d_1 \dots d_{k'}, i) \in S$$

Ahora que se ha demostrado la propiedad, se puede concluir sobre en tamaño de Q^{det}.
Sea $S, S' \in Q^{det}$, si $(w, M) \in S$ y $(w, M) \in S'$ entonces $S = S'$. Esto nos dice que los conjuntos están completamente determinados por el máximo, es decir, si tienen el mismo máximo, son el mismo conjunto.
Con lo anterior, se puede concluir que existe una función inyectiva $f : Q^{det} \to Q$, por lo tanto, el tamaño de Q^{det} no puede ser mayor que el tamaño de Q, que efectivamente es $n+m+1$, por lo tanto $|Q^{det}| \leq n+m+1$.
Notar que cuando la palabra u o la palabra v tienen secciones en común, la determinización puede tener menos estados, por lo que hay un $\leq$ y no directamente la igualdad, como en la demostración vista en clases.