

AUXILIAR #2 - AUTÓMATAS APILADORES Y GRAMÁTICAS LIBRES DE CONTEXTO

1 de abril de 2021 - Bernardo Subercaseaux

Una gramática libre de contexto (GLC) es una 4-tupla (V, Σ, R, S) donde V es un conjunto finito de *variables*. Σ es un alfabeto disjunto de V , cuyos símbolos llamamos *terminales*. R es un conjunto finito de reglas de la forma $A \rightarrow \alpha$, (se lee « A produce α »), con $A \in V, \alpha \in (V \cup \Sigma)^*$. S es una variable especial llamada *de partida*. Por ejemplo, considere la gramática $G = (\{A\}, \{a, b\}, \{S \rightarrow A, A \rightarrow ab, A \rightarrow ba, A \rightarrow AA\}, S)$. Diremos que G produce el string $abba$ ya que existe una secuencia de reglas de producción que comienza en S y termina en $abba$. En este caso, $S \rightarrow A \rightarrow AA \rightarrow abA \rightarrow abba$.

En general, dada una gramática GLC G , definimos L_G como el lenguaje de todas las palabras que produce G .

Problema 1. (★★) Construya GLCs para los siguientes lenguajes:

- $L = \{w \in \{(,)\}^* \mid w \text{ está bien parentetizado}\}$
- $L = \{w \in \{a, b\}^* \mid w \text{ es un palíndromo}\}$
- $L = \{w \in \{0, 1\}^* \mid \exists n \in \mathbb{N}, w = 0^n 1^n\}$
- $L = \{w \in \{a, b\}^* \mid w \text{ tiene la misma cantidad de a's que de b's}\}$
- **(Propuesto)** $L = \{w \in \{a, b\}^* \mid w \text{ tiene el doble de a's que de b's}\}$

Un autómata apilador (AP) es similar a un autómata finito no determinista con ϵ -transiciones, pero cuenta además con una estructura de datos llamada *pila* o *stack*. Un stack permite dos operaciones: PUSH(s), que pone el símbolo s en el *tope*, y PULL() que remueve el símbolo que está en el tope.

Formalmente, un AP es una 6-tupla $(Q, \Sigma, \Gamma, \delta, s, F)$. Las diferencias con los autómatas finitos corrientes son dos; Primero, los AP poseen Γ , un alfabeto de símbolos para el stack, que no necesariamente coincide con Σ . Segundo, la función de transición ahora es de la forma

$$\delta : Q \times \Sigma_\epsilon \times \Gamma_\epsilon \rightarrow 2^{Q \times \Gamma_\epsilon}$$

Es decir, al leer un símbolo del input (quizá ϵ), y el símbolo al tope del stack (quizá ϵ), se cambia de estado, se remueve el símbolo leído del tope del stack y se puede poner un nuevo símbolo al tope del stack. El nuevo símbolo a poner puede ser el mismo que se ha removido (es decir, no se modifica el stack), uno distinto, o ϵ .

Problema 2. (★★) Construya APs para los siguientes lenguajes, y prográmelos en JFLAP:

- $L = \{w \in \{(,)\}^* \mid w \text{ está bien parentetizado}\}$
- $L = \{w \in \{0, 1\}^* \mid \exists n \in \mathbb{N}, w = 0^n 1^n\}$
- **(Propuesto)** $L = \{w \in \{a, b\}^* \mid w \text{ tiene el doble de a's que de b's}\}$

Problema 3. (★★) Muestre que para todo lenguaje regular L existe una GLC G tal que $L_G = L$. Es decir, que todos los lenguajes regulares se pueden producir con GLCs.

Problema 4. (★★★) Muestre la equivalencia de poder expresivo entre GLCs y autómatas apiladores. Más precisamente:

- Demuestre que para todo autómata apilador A , existe una GLC G_A tal que $L_A = L_{G_A}$. **Nota:** por tiempo no alcanzamos a realizar esto durante la auxiliar.
- **(Propuesto)** Demuestre que para toda GLC G , existe un autómata apilador A_G tal que $L_G = L_{A_G}$.