Parseo y Generación de Código – 2$^{\text{do}}$ semestre 2019
Licenciatura en Informática con Orientación en Desarrollo de Software
Universidad Nacional de Quilmes

Primer parcial

Nota: este parcial es a libro abierto. Se permite tener cualquier material manuscrito o impreso, pero no se permite el uso de dispositivos electrónicos. El parcial se califica con una nota numérica de 1 a 10. Se requiere ≥ 4 en ambos parciales para aprobar la materia. Para promocionar se requiere nota ≥ 6 en ambos parciales y promedio ≥ 7.

Ejercicio 1. La siguiente gramática $G_1 = (\{P, C, E\}, \{\mathbf{id}, \mathtt{while}, :=, +, \mathbf{\{}, \mathbf{\}}\}, \mathcal{P}, P)$ describe la sintaxis de los programas (P), comandos (C) y expresiones (E) de un lenguaje de programación. El conjunto $\mathcal{P}$ de producciones está dado por:

$$\begin{array}{l} P \to \epsilon \mid PC \\ C \to \mathbf{id} := E \ \mid \mathtt{while}\ E\ P \mid \mathbf{\{} P \mathbf{\}} \\ E \to \mathbf{id} \mid \mathbf{id} + \mathbf{id} \end{array}$$

a. Demostrar que G_1 es ambigua.

b. Proponer una gramática G_2 que genere el mismo lenguaje que G_1 pero que sea LL(1). Armar la tabla LL(1) y mostrar que no tiene conflictos.

Ejercicio 2. Dada la siguiente expresión regular en el alfabeto $\{a, b, c\}$:

$$R = (a|b)^*c(a|b)^*c$$

a. Proponer una gramática SLR que genere el lenguaje $\mathcal{L}(R)$. Armar la tabla SLR.

b. Analizar sintácticamente la cadena $acbc$ usando la tabla SLR armada.

Ejercicio 3. Sean $L_1 = \{\alpha \in \{a, b\}^* \mid |\alpha|_a > 2\}$ y $L_2 = \mathcal{L}(a(a|b)^*b)$. Dar una expresión regular para el lenguaje $L_1^c \cap L_2^c$. Recordar que L^c denota el complemento de L, es decir $L^c = \{\alpha \in \{a, b\}^* \mid \alpha \notin L\}$.

Ejercicio 4. Si $L \subseteq \Sigma^*$ es un lenguaje cualquiera, llamamos $L^{[2]}$ al lenguaje que resulta de concatenar un número par de palabras de L:

$$L^{[2]} = \{\alpha_1 \dots \alpha_n \mid n \geq 0 \text{ es par y para todo } 1 \leq i \leq n \text{ se tiene } \alpha_i \in L\}$$

Por ejemplo, si $L = \{aa, bc\}$ entonces $aabcbcaa \in L^{[2]}$ pero $aabcbc \notin L^{[2]}$. Más en general, llamamos $L^{[k]}$ al lenguaje que resulta de concatenar un número de palabras de L que sea múltiplo de k.

$$L^{[k]} = \{\alpha_1 \dots \alpha_n \mid n \geq 0 \text{ es un múltiplo de } k \text{ y para todo } 1 \leq i \leq n \text{ se tiene } \alpha_i \in L\}$$

Demostrar que si L es un lenguaje regular, entonces para todo $k \geq 2$ se tiene que $L^{[k+1]} \setminus L^{[k]}$ es regular. Recordar que $A \setminus B$ denota la diferencia de conjuntos, es decir $A \setminus B = \{x \mid x \in A \land x \notin B\}$.

Justificar todas las respuestas.