Parseo y Generación de Código – 2$^{\text{do}}$ semestre 2017
Licenciatura en Informática con Orientación en Desarrollo de Software
Universidad Nacional de Quilmes

Práctica 3
Normalización de gramáticas

Ejercicio 1. Dar una gramática no ambigua en el alfabeto $\Sigma = \{\mathbf{if}, \mathbf{then}, \mathbf{else}, \mathbf{exp}, \mathbf{cmd}\}$ que genere el mismo lenguaje que la siguiente gramática (que ya sabemos ambigua, *cf.* Ejercicio 3 de la práctica 1):

$$\begin{array}{rcl} S & \to & \mathbf{if}\ E\ \mathbf{then}\ S\ \mathbf{else}\ S \mid \mathbf{if}\ E\ \mathbf{then}\ S \mid \mathbf{cmd} \\ E & \to & \mathbf{exp} \end{array}$$

Para eliminar la ambigüedad, decidir el problema del *dangling else* optando siempre por asociar cada **else** con el **if** que se encuentre lo más a la derecha posible.

Ejercicio 2. Dada la siguiente gramática G en el alfabeto $\Sigma = \{\mathbf{n}, +, -\}$:

$$E \quad \to \quad \mathbf{n} \mid E + E \mid -E$$

1. Dar una gramática G_1 sin recursión a izquierda que genere el mismo lenguaje que G.

2. Demostrar que G es ambigua exhibiendo dos árboles de derivación distintos para $-\mathbf{n} + \mathbf{n}$.

3. Dar una gramática G_2 no ambigua que genere el mismo lenguaje que G. Para eliminar la ambigüedad, suponer que el operador binario $+$ es asociativo a izquierda, y que el operador unario $-$ tiene mayor precedencia que $+$.

Ejercicio 3. Dada la siguiente gramática G que describe expresiones regulares, usando $+$ en lugar de $\mid$ para no confundir el operador de expresiones regulares con la barra $\mid$ del metalenguaje:

$$R \quad \to \quad \mathbf{a} \mid R + R \mid R \cdot R \mid R^{\star} \mid (R)$$

1. Dar una gramática G_1 no ambigua que genere el mismo lenguaje que G. Asumir que las relaciones de precedencia son las usuales, es decir que el operador $+$ tiene mayor precedencia que el operador $\cdot$, y que este tiene mayor precedencia que el operador de clausura.

2. A partir de la gramática G_1, dar una gramática G_2 sin recursión a izquierda y no ambigua que genere el mismo lenguaje.

Ejercicio 4. Usando el método visto en la materia, eliminar las producciones ϵ de las siguientes gramáticas:

1. $G_1 = (\{S, B\}, \{a, b\}, P, S)$ donde P son las producciones:

$$\begin{array}{rcl} S & \rightarrow & aB \\ B & \rightarrow & \epsilon \mid bB \end{array}$$

2. $G_2 = (\{S, A, B, C\}, \{a, b, f, g\}, P, S)$ donde P son las producciones:

$$\begin{array}{rcl} S & \rightarrow & fA \mid gAA \\ A & \rightarrow & a \mid BC \\ B & \rightarrow & b \mid \epsilon \\ C & \rightarrow & \epsilon \end{array}$$

Ejercicio 5. Usando el método visto en la materia, eliminar los ciclos de las siguientes gramáticas:

1. $G_1 = (\{S, A, B\}, \{a, b\}, P, S)$ donde P son las producciones:

$$\begin{array}{rcl} S & \rightarrow & AB \\ A & \rightarrow & \epsilon \\ B & \rightarrow & AB \mid a \mid b \end{array}$$

2. $G_2 = (\{S, A, B, C, D\}, \{a, b\}, P, S)$ donde P son las producciones:

$$\begin{array}{rcl} S & \rightarrow & aB \\ A & \rightarrow & C \mid BC \mid a \\ B & \rightarrow & C \mid b \\ C & \rightarrow & D \\ D & \rightarrow & B \mid A \end{array}$$

Ejercicio 6. Usando el método visto en la materia, eliminar la recursión a izquierda de las siguientes gramáticas:

1. $G_1 = (\{S, L, M, N\}, \{(, a, \circ, \bullet,)\}, P, S)$ donde P son las producciones:

$$\begin{array}{rcl} S & \rightarrow & (L) \\ L & \rightarrow & \epsilon \mid M \\ M & \rightarrow & a \mid Na \\ N & \rightarrow & M \bullet \mid M \circ \end{array}$$

2. $G_2 = (\{S, A, B, C, D\}, \{a, b, c, d, f\}, P, S)$ donde P son las producciones:

$$\begin{array}{rcl} S & \rightarrow & A \\ A & \rightarrow & AAa \mid Bb \mid c \mid d \\ B & \rightarrow & Af \mid f \end{array}$$