

# Teoría de Lenguajes - Segundo Parcial

Primer cuatrimestre de 2020

Apagar los celulares.

Hacer cada ejercicio en hojas separadas.

Poner nombre, número de orden y número de página en cada ejercicio.

Justificar todas las respuestas.

El examen es a libro abierto.

Se aprueba con 65 ó más puntos.

1. (34 pts) Dada la gramática:  $G = \langle \{S, A, B, C, D\}, \{2, 3\}, S, P \rangle$ , con  $P$ :

$$\begin{aligned} S &\rightarrow 3 \mid A \mid 3A \\ A &\rightarrow 2C \mid C \\ B &\rightarrow \lambda \mid B2 \mid SS \\ C &\rightarrow 2 \mid S \\ D &\rightarrow DB \mid A3 \end{aligned}$$

- a) Aplicar la eliminación de símbolos de modo que queden eliminadas producciones innecesarias sin modificar ni agregar otras, y de modo que se preserve el lenguaje  $L(G)$ .
- b) Determinar si la gramática resultante de (a) es SLR, indicando todos los conflictos que pudiera haber asociados a la tabla.
- c) Determinar si la gramática resultante de (a) es LR(0).
2. (33 pts) Considerar un lenguaje de programación reducido que usa sólo el tipo entero, en el que un programa es un secuencias de 0 o más sentencias separadas estas por `;`, donde cada sentencia puede ser:
- asignación (múltiple) de un número a una o más variables, de la forma  $var = var = \dots = var = num$
  - asignación (múltiple) de una variable a otras, de la forma  $var = var = \dots = var$
  - incremento del valor de una variable, de la forma  $var++$
  - impresión del valor de una variable, de la forma  $pr\ var$

El token *num* tiene un atributo numérico que representa un valor entero. Asumir que las variables arrancan con valor 0.

Se pide definir una GLC sin símbolos inútiles, utilizando los atributos que se considere necesarios, de modo que quede definida una TDS que permita que las asignaciones e impresiones funcionen del modo usual, y que además al final de la ejecución de la secuencia se imprima el máximo valor que haya sido impreso anteriormente (si no se imprimió nada antes, se imprimirá un valor arbitrario o nada). Como código dentro de la TDS se pueden utilizar sentencias de C habituales y diccionarios accesibles globalmente (no otras funciones externas ni bibliotecas).

Ejemplo:  $b = c = 5; a = 3; b = a; b++; pr\ c; pr\ b$  es una cadena válida, que al ejecutarse imprime  $5\ 4\ 5$ .

3. (33 pts) Sea  $G$  la gramática extendida  $\langle \{S, B\}, \{a, b, c, ' , d\}, S, P \rangle$ , con  $P$ :

$$\begin{aligned} S &\rightarrow a(c?aB \mid aB)^*d(c?)^+c \\ B &\rightarrow (b,)^*b \end{aligned}$$

- a) Determinar si  $G$  es ELL(1). Si no lo es, dar otra gramática  $G'$  que sí lo sea y tal que  $L(G) = L(G')$ .
- b) Dar el parser iterativo-recursivo para  $G$  si  $G$  fuera ELL(1) o para  $G'$  en otro caso.