

Tarea I

(10 puntos)

A continuación encontrará 2 preguntas, cada una compuesta de diferentes sub-preguntas. El valor de cada pregunta (y sub-pregunta) estará expresado entre paréntesis al inicio de las mismas.

En aquellas preguntas donde se le pida ejecutar un algoritmo o procesar una entrada, incluya los pasos relevantes de la ejecución del mismo con los cuales usted pudo alcanzar su conclusión. Sea lo más detallado y preciso posible en sus razonamientos y procedimientos.

En aquellas preguntas donde se le pida implementar un programa, mantenga su código en un repositorio `git` remoto (preferiblemente `Github`) y coloque un enlace al mismo en lugar de su respuesta. Todo su código debe ser legible y estar debidamente documentado (incluyendo instrucciones para su correcta ejecución).

La entrega se realizará por correo electrónico a `rmonascal@gmail.com` hasta las 11:59pm. VET del Jueves, 13 de Febrero de 2025.

1. (3 puntos) Considere el lenguaje JSON (JavaScript Object Notation). Se desea que:
 - a) (1 punto) Plantee una gramática libre de contexto que genere frases correctas en JSON.
 - b) (2 puntos) Implemente un reconocedor recursivo descendente para su gramática, usando una herramienta basada en `parsec` (información disponible en: [https://en.wikipedia.org/wiki/Parsec_\(parser\)](https://en.wikipedia.org/wiki/Parsec_(parser))). Puede utilizar el lenguaje de su elección, siempre y cuando tenga una implementación oficial de `parsec` disponible.
2. (7 puntos) Sea una gramática $IFT = (\{n, +, \text{if}, \text{then}, \text{else}\}, \{E\}, P, E)$, con P definido de la siguiente manera:

$$\begin{array}{lcl}
 E & \rightarrow & \text{if } E \text{ then } E \text{ else } E \\
 & | & \text{if } E \text{ then } E \\
 & | & E + E \\
 & | & n
 \end{array}$$

- a) (2 puntos) Aumente la gramática con un nuevo símbolo inicial no recursivo S , construya la máquina característica LR(1) y diga si existen conflictos en la misma.
- b) (2 puntos) Tome en consideración las siguientes reglas:

Regla	Ejemplo
$+$ asocia a izquierda.	$E + E + E = (E + E) + E$
else se asocia al if más interno.	$\text{if } E \text{ then if } E \text{ then } E \text{ else } E = \text{if } E \text{ then } (\text{if } E \text{ then } E \text{ else } E)$
if tiene mayor precedencia que $+$.	$E + \text{if } E \text{ then } E \text{ else } E + E = E + (\text{if } E \text{ then } E \text{ else } E) + E$

En caso de haber conflictos en el autómata de prefijos viables LR(1), diga cómo resolvería los conflictos (seleccionando una de las acciones que conforma dicho conflicto), de tal forma que las reglas anteriores sean satisfechas.

- c) (2 puntos) A partir de las respuestas anteriores, construya la máquina característica LALR(1) y diga si existen conflictos en la misma. En caso de existir, explique cómo los resolvería (seleccionando una de las acciones que conforma dicho conflicto), con las mismas reglas de la pregunta anterior.
- d) (1 punto) Use la máquina LALR(1) para procesar la frase:

`if n + n then if n then n else n + n`