Universidad del Valle de Guatemala
Facultad de Ingeniería
Ingeniería en Ciencia de la Computación y Tecnologías de la Información

Laboratorio No. 7

Para los siguientes incisos, siga estas instrucciones:

- Cree un repositorio **privado** en GitHub.com para alojar el código correspondiente para este laboratorio.
- Añada los usuarios de GitHub de su catedrático y de su(s) auxiliar(es) con rol de *collaborator*. Estos se encuentran en la sección de información del curso en Canvas.
- Coloque todas las instrucciones que considere pertinentes en un archivo denominado README.md en la raíz del repositorio.
- Para los incisos que no involucren la generación de código, cree una carpeta por separado en su repositorio y coloque las respuestas en un documento con formato PDF
- Grabe un video de no más de 10 minutos donde muestre la ejecución de sus programas para el ejercicio 2. Súbalo a YouTube como video no listado y adjúntelo en el README de su repositorio.

Ejercicio No. 1 (50%) – Para las siguientes dos CFGs:

1. $S \rightarrow 0A0 \mid 1B1 \mid BB$
   $A \rightarrow C$
   $B \rightarrow S \mid A$
   $C \rightarrow S \mid \varepsilon$

2. $S \rightarrow aAa \mid bBb \mid \varepsilon$
   $A \rightarrow C \mid a$
   $B \rightarrow C \mid b$
   $C \rightarrow CDE \mid \varepsilon$
   $D \rightarrow A \mid B \mid ab$

a) Elimine las producciones-ε.
b) Elimine cualquier producción unaria de la gramática resultante.
c) Elimine cualquier símbolo inútil de la gramática resultante.
   a. Remueva símbolos que no producen.
   b. Remueva símbolos no alcanzables.
d) Coloque la CFG resultante en la Forma Normal de Chomsky (CNF).

Muestre todo su procedimiento, como vimos en clase.

Ejercicio No. 2 (50%) – Seleccione un lenguaje de programación de alto nivel y cree un programa que realice las siguientes operaciones relacionadas a la simplificación de gramáticas:

- Su programa deberá cargar archivos de texto que contengan gramáticas. Para este laboratorio, cree dos archivos distintos, uno por cada una de las gramáticas del Ejercicio No. 1.
  - Cada línea representa una producción en su gramática. Puede tener más de una producción por línea si estas están separadas por un operador OR "|".
  - Para este laboratorio, la convención será que las letras individuales en mayúscula son no-terminales y las letras individuales en minúscula son terminales. Más

adelante, en proyectos subsiguientes esta convención puede verse afectada por la introducción de palabras que representan a los terminales o no terminales.
    - Su programa deberá interpretar correctamente que cada línea del archivo esté bien escrita, de lo contrario la ejecución debe detenerse. Para ello, utilice el trabajo empleado en el Proyecto 1 de este curso para validar que cada línea está bien escrita. Por ejemplo, cree una regex que acepte producciones para gramáticas, es decir su regex debe validar que pueda escribir expresiones como $S \rightarrow 0A0 \mid 1B1 \mid BB$, significando un texto que comience con un símbolo en mayúscula, seguido de una flecha, seguido de cuerpos de producción, que pueden estar separados por un OR.
        - En el video de demostración, cambie ligeramente las producciones para introducir errores y que se vea reflejada esta validación de sus gramáticas.
        - Una regex para aceptar producciones como la anterior podría verse algo como (es un ejemplo): $[A-Z](\backslash n) \rightarrow ((\backslash n)([A-Za-z0-9])^{*}(\backslash n\backslash \mid)^{*})^{+} \mid \backslash \varepsilon$
- Luego de interpretar las producciones dentro de los archivos de texto, cree un algoritmo en su programa que elimine producciones-ε. Deberá mostrar en pantalla la ejecución de los pasos de su algoritmo para remover las producciones de este tipo y deberá mostrar el resultado de la gramática sin producciones-ε.
    - Recuerde que para remover producciones-ε deberá:
        - Encontrar símbolos y producciones anulables.
        - Encontrar las nuevas producciones de la gramática tomando en cuenta todos los 2^m casos posibles para formular producciones con base en aquellas que tengan m símbolos anulables.