



Laboratorio E

Descripción

Este laboratorio consiste en iniciar el desarrollo de un generador de analizadores sintácticos basado en YAPar, el cual, tomando como entrada una especificación de componentes sintácticos, generará un analizador sintáctico que será capaz de analizar sintácticamente un flujo de tokens proporcionado por un analizador léxico, el cual será generado por su laboratorio D.

Su programa deberá de leer el archivo escrito en lenguaje YAPar, interpretarlo, identificar la gramática, generar el autómata LR(0) y mostrar una representación visual de este, incluyendo los elementos correspondientes dentro de cada conjunto.

La sintaxis de un archivo escrito en lenguaje YAPar se especifica en el documento **Consideraciones de YAPar**.

Objetivos

General

- Desarrollar un generador de analizadores sintácticos basado en YAPar.

Específicos

- Aplicar los conceptos de análisis sintáctico en la construcción de un generador de analizadores sintácticos.
- Analizar sintácticamente tokens.
- Detectar errores sintácticos y reportarlos de forma inteligible.
- Desarrollar los algoritmos: construcción de un autómata LR(0), función cerradura, función ir a, función primero y función siguiente.

Especificación del funcionamiento del generador de analizadores sintácticos

Entrada

- Un archivo que contiene la especificación de un analizador léxico a generar, escrita en el lenguaje YALex.
- Un archivo que contiene la especificación de un analizador sintáctico a generar, escrita en el lenguaje YAPar.

Salida

- El diagrama de transición de estados del autómata LR(0), construido a partir de la especificación de componentes sintácticos proporcionada.

Consideraciones y requerimientos

- Para poder elaborar de este laboratorio, debe haber completado en su totalidad los laboratorios anteriores.
- Tanto el lenguaje de programación utilizado para construir el generador de analizadores sintácticos, como el lenguaje de programación de los analizadores sintácticos generados, quedan a su elección.

- Deberá escribir los archivos de prueba que se utilizarán en la calificación; al menos tres: uno con complejidad baja, otro con complejidad media y otro con complejidad alta (el incumplimiento de esto se penalizará restando un punto a la nota obtenida).
- El software deberá contar con una interfaz gráfica amigable y estética (el incumplimiento de esto se penalizará restando un punto a la nota obtenida).
- El uso de librerías para expresiones regulares está prohibido (el incumplimiento de esto se penalizará restando tres puntos a la nota obtenida).
- Deberá mostraren en alguna sección de su interfaz gráfica el resultado de las funciones primero y siguiente.
- Se sugiere programar la representación gráfica del autómata LR(0) haciendo uso de librerías ya existentes.

Rúbrica de calificación

Funcionalidad a evaluar	Ponderación
Interpretación de gramática siguiendo todos los lineamientos del archivo Consideraciones de YAPar .	5 puntos
Validación de que tokens definidos en archivo YAPar se encuentren también definidos en archivo YALex (de no ser así, mostrar un error).	3 puntos
Utilizando el archivo con complejidad baja, visualización del autómata LR(0).	2 puntos
Modificación del archivo con complejidad baja y utilizando dicho archivo, visualización del autómata LR(0).	2 puntos
Utilizando el archivo con complejidad media, visualización del autómata LR(0).	2 puntos
Modificación del archivo con complejidad media y utilizando dicho archivo, visualización del autómata LR(0).	2 puntos
Utilizando el archivo con complejidad alta, visualización del autómata LR(0).	2 puntos
Modificación del archivo con complejidad alta y utilizando dicho archivo, visualización del autómata LR(0).	2 puntos
Cálculo de funciones primero y siguiente.	5 puntos