

Laboratorio F

Descripción

Este laboratorio consiste en finalizar la implementación de un Generador de Analizadores Sintácticos, **YAPar**. Este deberá tomar como entrada una especificación de componentes sintácticos, i.e. una gramática que defina a un lenguaje regular, siguiendo la sintaxis correcta de un archivo en lenguaje YAPar. Asimismo, tomará como entrada también los tokens e información de la tabla de símbolos resultante de la previa ejecución de su Generador de Analizadores Léxicos, **YALex**.

Para esta última fase de su Generador de Analizadores Sintácticos, su programa deberá de utilizar el resultado del Laboratorio E, es decir, el autómata LR(0) generado y con este, deberá de construir una tabla de parseo SLR(1). Con ello, su programa deberá de generar un archivo de código fuente en el lenguaje de programación de su elección, el cual deberá de poderse ejecutar y proveer como entrada otro archivo por separado con el mismo contenido de cadenas evaluadas anteriormente por su Generador de Analizadores Léxicos.

De esta forma, ahora su Generador de Analizadores Sintácticos, generará un archivo ejecutable que usted puede usar para *parsear* el archivo de entrada con cadenas y de esta forma analizarlo desde una perspectiva sintáctica, informando si este archivo es sintácticamente correcto o no. Por ello, deberá de implementar el algoritmo de parsing SLR(1) sobre esta entrada para poder determinar lo solicitado.

El archivo **Consideraciones de YAPar** explica más a fondo la estructura de un archivo en YAPar.

Objetivos

- **Generales**

- Implementar un Generador de Analizadores Sintácticos [YAPar]

- **Específicos**

- Aplicar la teoría de Analizadores Sintácticos en la construcción de una herramienta de software generadora de dichos componentes en un lenguaje de programación a su elección.
- Implementar una herramienta de software que sea capaz de generar Analizadores Sintácticos funcionales basado en la especificación de componentes sintácticos (Gramática SLR) y la recolección de tokens producidos por su generador de Analizadores Léxicos.
- Implementar la construcción de una tabla de parseo SLR(1) y ejecutar parsing sobre un archivo de cadenas provisto.

Especificación del funcionamiento del generador de analizadores sintácticos.

- **Entrada**

- Un archivo que contiene la especificación del analizador léxico a generar. Esta especificación se encuentra escrita en el lenguaje YALex. Producirá tokens que serán luego interpretados por su implementación de YAPar.
- Un archivo que contiene la especificación del analizador sintáctico a generar. Esta especificación se encuentra escrita en el lenguaje YAPar.
- Un sólo archivo que contiene cadenas de entrada para ambos Generadores.

- **Salida**

- Resultado en pantalla sobre la ejecución de parsing, utilizando la tabla de parseo SLR(1) generada, sobre un archivo de entrada con cadenas a ser evaluadas.

Consideraciones y requerimientos

- Para la elaboración de este Laboratorio, debe de tener entrega completa de los laboratorios A, B, C y D.
- Su programa debe escanear el archivo de YAPar, interpretarlo y con ello, identificar la gramática que se describe en esta especificación, luego usando esta gramática, deberá construir el autómata LR(0) correspondiente a esta gramática.
- Se le proveerán diversos archivos .yalp para que ingrese en su programa como especificaciones para su generador de analizadores sintácticos.
- La generación visual del autómata LR(0) se refiere a una representación en pantalla que simplifique la comprensión sobre la generación efectuada por su programa. Por lo tanto, esta puede ser llevada a cabo con librerías de código para generar un autómata visual utilizando elementos gráficos y generando un archivo de imagen, por ejemplo. Otra implementación válida sería la generación gráfica de un autómata por medio de alguna página web o similar. La representación gráfica más simple sería colocar en pantalla un listado comprensivo de nodos del autómata, con sus correspondientes elementos LR(0) y un listado de las transiciones que este mismo tiene hacia otros nodos.
- En caso la implementación del autómata LR(0) no pueda ser observada de manera adecuada, es decir, la implementación no funciona correctamente o no se presenta en pantalla el mismo, considere mostrar cada una de las fases previas a la generación del autómata LR(0), correspondientes a los rubros en la tabla de ponderación colocada más abajo. Esto facilitará la calificación de cada uno de esos rubros para validar que en efecto cada uno de ellos haya sido desarrollado correctamente.
- Diseñe una arquitectura genérica para su solución de software:
 - Reutilice código al crear objetos y estructuras de datos que le permitan construir su autómata y en el cual pueda diseñar la implementación de métodos/funciones asociadas para los cálculos correspondientes.
 - Planee a futuro, escriba código genérico y parametrizable.

Ponderación

Este laboratorio tiene un valor total de **30 puntos netos**. Su distribución es la siguiente:

Item a evaluar	Ponderación
Construcción de tabla de parseo SLR(1), incluyendo todas sus partes correspondientes: ACTION table, GOTO table y estados, generada a partir del autómata LR(0).	15 puntos
Implementación de algoritmo de parsing SLR(1) para la evaluación del archivo de entrada con cadenas.	10 puntos
Indicación de errores sintácticos como resultado de la aplicación del algoritmo de parseo sobre el archivo de cadenas provisto.	3 puntos
Implementación para la Indicación de errores gramaticales al evaluar las entradas en la tabla de parseo SLR(1), identificando conflictos en la tabla para gramáticas que no son SLR.	2 puntos