

Generador de Analizador Sintáctico Descendente con Retroceso (ASDR)

1. Objetivo

El trabajo consiste en programar en C un **Generador** de Analizador Sintáctico Descendente con Retroceso (ASDR). Dada una gramática libre de contexto, **sin recursividad a izquierda**, el programa deberá generar un ASDR que estará escrito en lenguaje C (ASDR.c). El ASDR podrá luego recibir una cadena de entrada y determinar si pertenece o no al lenguaje generado por la gramática, mostrando además la derivación que llevó a dicha cadena.

2. Descripción del funcionamiento esperado

2.1. Programa principal. (Generador de ASDR)

El generador recibe como entrada un archivo de extensión .gr, donde está la especificación de una sola gramática que se asume libre de contexto y sin recursividad a izquierda. La sintaxis de ejecución del generador es:

genASDR nombearchivogramatica

El generador obtiene como salida un archivo ASDR.c que será el código fuente del Analizador Sintáctico Descendente con Retroceso implementado a través de procedimientos.

El ASDR deberá tener una función Main, una función por cada no terminal de la gramática y una función Procesar. Se sugiere como heurística la siguiente: mientras el método intenta diferentes derivaciones, si el conjunto de hojas del árbol es mayor a la longitud de la cadena a analizar entonces se aplica el retroceso.

Un archivo de gramática (.gr) será un archivo en el que se especificarán las gramáticas de la siguiente forma: NombreDeLaGramatica = ({SimbolosNoTerminales}, {SimbolosTerminales}, SimboloInicial, {Producciones})

Consideraciones:

- Los SimbolosNoTerminales son letras en mayúscula separadas por comas.
- Los SimbolosTerminales son letras en minúscula separadas por comas.
- Para el símbolo lambda se usará el caracter \ (barra invertida).
- Se debe contemplar que los caracteres del archivo pueden (o no) estar separados por espacios, tabuladores o fines de línea.
- Se recibe una sola gramática por archivo.
- El nombre de la gramática empieza por una letra. (después puede seguir cualquier símbolo, sin espacios)

El siguiente podría ser el contenido de un archivo válido:

$G_1 = (\{A, B, C\}, \{a, b, c\}, A, \{A \rightarrow aBC | cB, B \rightarrow aA | b, C \rightarrow c | \})$

Dada la gramática de entrada, deberá obtener como salida un programa en C (ASDR.c) que será el Analizador Sintáctico Descendente con Retroceso.

2.2. Analizador Sintáctico Descendente con Retroceso.

El analizador recibe como entrada una palabra que podría pertenecer al lenguaje es decir, una cadena formada sólo por letras minúsculas. La sintaxis de ejecución del analizador es:

ASDR *palabra*

El programa ASDR deberá devolver si la cadena pertenece al lenguaje o no, y además o bien el árbol de derivación que dió lugar a esa cadena o bien el conjunto de producciones utilizadas para derivar la cadena o bien el conjunto de las formas sentenciales izquierdas desde el símbolo distinguido hasta la cadena. (Cualquiera de las tres opciones o todas si quieren).

3. Alternativa: Compilador de un lenguaje de programación simple.

Cómo alternativa, se acepta la implementación de una gramática con la definición de un lenguaje procedural simple que implemente como mínimo:

- Funciones (con/sin recursividad).
- Tipos de datos (char y entero como mínimo).

El compilador deberá generar en base a un programa, que es una palabra de la gramática del lenguaje, verificar su correcta sintaxis y generar una salida en .c, .java, java bytecode, .asm o binario puro para poder ejecutarse sobre la arquitectura destino. Finalmente debe poder escribirse un programa en este lenguaje que permita calcular el factorial de un número ingresado (entrada) y genere el resultado (salida). Se valorará positivamente la rapidez en la ejecución del compilador, la rapidez en la ejecución del programa y cualquier optimización del código generado aplicada.

4. Material a entregar

- Códigos fuente: Makefile, código Lex (sin compilar), y códigos C.
- Archivo ejecutable para Linux/Mac OS.
- Un readme explicando cómo compilar y ejecutar el programa.
- Archivos de ejemplo diferentes a las provistas por la cátedra.
- Un informe que contenga, en este orden:
 - Carátula
 - Índice
 - Consideraciones realizadas (no previstas en el enunciado).
 - Descripción del desarrollo del TP.
 - Dificultades encontradas en el desarrollo del TP.
 - Futuras extensiones, con una breve descripción de la complejidad de cada una.

5. Grupos

El trabajo se realizará en grupos, integrados de la misma manera que para el TP1.

6. Fecha de entrega

El material a entregar debe ser enviado por mail, en un archivo tipo .zip a la cuenta rramele@gmail.com antes de finalizado el día 23 de Noviembre.