

## LABORATORIO N°2

### *Lenguajes Libres del Contexto - Autómata Apilador*

**Fecha de envío:** Domingo 19 de mayo, 23:55 hrs.

**Modalidad:** Trabajo en grupos de **máximo** dos personas.

#### I. Objetivo.

El objetivo de este laboratorio es evaluar su capacidad para: Diseñar e implementar un autómata apilador.

#### II. Enunciado.

Diseñe y programe un autómata apilador que reciba como entrada un archivo de texto con el código fuente de un programa en C, e indique por pantalla al usuario si los paréntesis del código fuente están bien balanceados, o no. En el caso de no estar bien balanceados, debe indicar la línea donde encontró el error, y qué paréntesis esperaba encontrar. Los paréntesis que debe revisar son:

- Paréntesis de llaves, que delimitan todo tipo de bloques de instrucciones: { }
- Paréntesis redondos, que delimitan los argumentos de operaciones matemáticas, lógicas, relacionales, y argumentos en las llamadas a funciones: ( )
- Paréntesis cuadrados, que delimitan el tamaño de cada dimensión que se definen para los arreglos de una o más dimensiones: [ ]

**IMPORTANTE:** Su programa **no debe revisar la sintaxis** del código fuente.

#### III.1 Interfaz.

Al ejecutar su programa, deberá presentar exactamente la interfaz de la figura 1. Primero, debe solicitar el nombre del archivo con el código fuente que se analizará (línea 1). El archivo debe ser ingresado con su extensión (que será siempre “.c”). Luego, deberá mostrar el resultado de su autómata apilador, señalando que los paréntesis están balanceados (línea 2) o que tiene un error de balance en la línea que fue detectado, y el paréntesis que se esperaba encontrar (línea 3). Finalmente, su programa debe dar la posibilidad de probar otro archivo (línea 5).

Línea	Texto
1	Ingrese nombre de archivo: Nombre de archivo incluyendo la extensión “.c”
2	Los paréntesis están balanceados.
3	Tiene un error de balance de paréntesis en la línea XX. Se esperaba un: {, }, (, ), [, ]
	Desea probar otro archivo S/N? <S o N, según desee el usuario>

*Figura 1: Estructura de la interfaz que debe poseer su programa.*



#### IV. Consideraciones en la revisión.

1. Debe implementar este trabajo en lenguaje C (no C++).
2. Debe construir **una función para cada tarea que realice su código**, cuidando los tipos de datos de las entradas y de las salidas.
3. Debe usar identificadores representativos para sus variables, parámetros de entrada y para sus funciones.
4. Debe comentar cada una de sus funciones, estructuras y tipos de datos que defina, indicando una descripción de la labor que lleva a cabo, sus entradas, y sus salidas.
5. Su código debe estar correctamente *indentado* (uso de sangrías para cada sub-bloque de instrucciones).
6. Puede trabajar con el IDE y compilador de C, que más le acomode. No obstante lo anterior, su código será revisado en entorno Windows.
7. **IMPORTANTE 1:** Como los autómatas apiladores son autómatas no determinísticos, el manejo de sus transiciones no se puede hacer a través de una matriz de transiciones. Por lo anterior, usted debe construir una función que tenga definida cada una de las transiciones  $\delta(p, a, \beta) = (q, \gamma)$  que requiere su autómata.
8. **IMPORTANTE 2:** Su programa debe trabajar explícitamente con una pila, realizando las operaciones de consulta del símbolo en el tope de la pila, de Push y de Pop. Decida usted si la representa mediante una representación estática o dinámica
9. **SINO CUMPLE CON LOS PUNTOS 7 Y 8 TENDRÁ INMEDIATAMENTE UN 1.0 EN ESTE LABORATORIO.**

#### V. Sobre la entrega, atrasos y faltas a la ética.

1. Debe subir su trabajo a la plataforma de unabvirtual.cl, en una casilla que se habilitará especialmente para esto.
2. Su trabajo debe ser subido por **SOLO UNO** de los integrantes del grupo.
3. Por cada 12 horas de atraso se descontará de su nota final **5 décimas**. Este descuento se aplicará desde el primer minuto de atraso.
4. Debe subir solo el archivo con el código fuente.
5. El nombre de este archivo debe ser Apellido Paterno y Primer Nombre de cada integrante del grupo. Ejemplo: Si Rosa González trabajó con Santiago Hernández, el archivo se debe llamar: **GonzalezRosaHernandezSantiago.c**
6. Si el programa no se puede ejecutar, tendrá la nota mínima: **1.0**. Si su programa funciona, se evaluará su ejecución, y también el código fuente (con los detalles solicitados en este documento).
7. Si existe sospecha de copia (con otros compañeros, o desde internet), será interrogado acerca de su trabajo, para aclarar dudas de su entendimiento y autoría. Si se confirma la sospecha, el trabajo será evaluado con nota **1.0**.