Analizador Sintáctico

BISON

Preparado por

Alexis Boza

Silvia Delgado

Instituto Tecnológico de Costa Rica

03/06/2013

Descripción del Problema

Para el curso de “Compiladores e Intérpretes” de la Escuela de Computación del Instituto Tecnológico de Costa Rica es necesario implementar un pequeño analizador sintáctico o parser para el lenguaje de etiquetas XHTML, el mismo deberá interactuar con el analizador léxico desarrollado como primer proyecto del curso. El parser deberá generar el árbol de análisis sintáctico según la gramática especificada, tomando en consideración que dicho árbol será utilizado posteriormente por un analizador semántico.

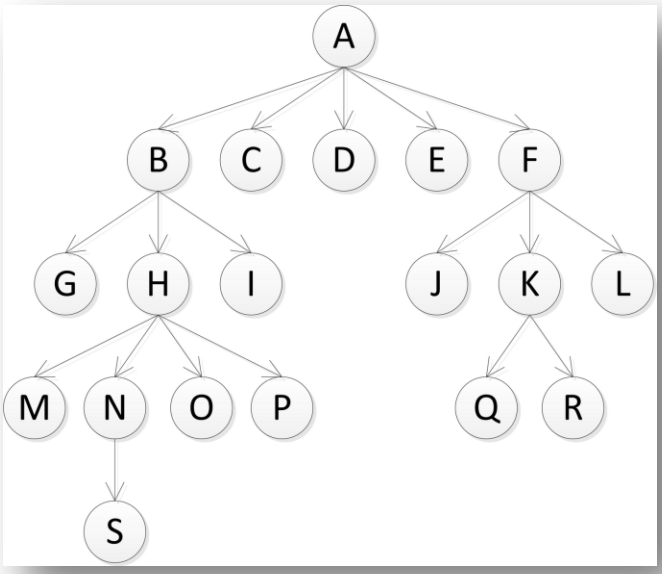
Además de generar el árbol de análisis sintáctico, deberá retornar al STDERR aquellos errores de estructura que violen las reglas gramaticales declaradas para el analizador. Según especificaciones brindadas con anterioridad el parser deberá funcionar solamente utilizando argumentos de línea de comandos.

El control del proyecto deberá ser llevado haciendo uso de la herramienta GITHUB.

Diseño del Programa

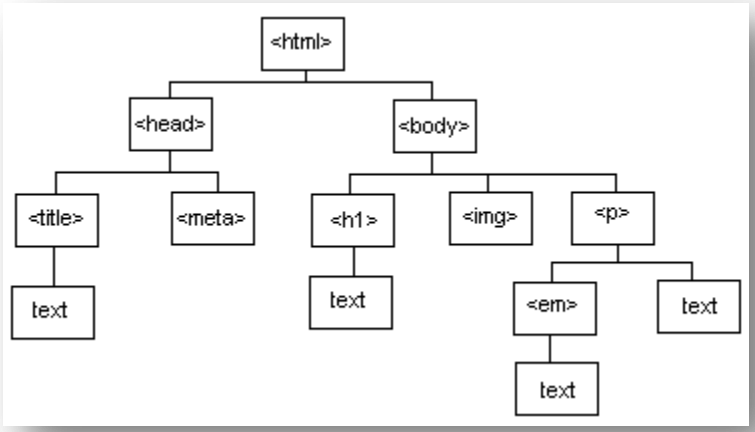
Árbol de análisis sintáctico

Para crear un árbol de análisis sintáctico que se adaptara a las necesidades de nuestro caso, se implementó un árbol N-ario en donde podríamos ingresar N cantidad de nodos para un mismo nodo.



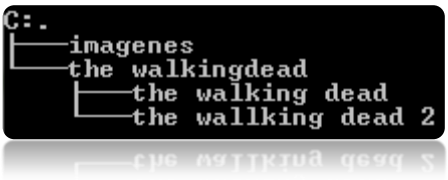
Ejemplo Árbol N-ario

La implementación del árbol puede ser visualizada en el archivo “tree.c”, el árbol parte de un nodo raíz el cual marca el inicio del documento HTML, hecho esto el árbol será construido en base a las etiquetas que son capturadas por el analizador léxico y que posteriormente el parser por medio de las reglas gramaticales corroborará su estructura.

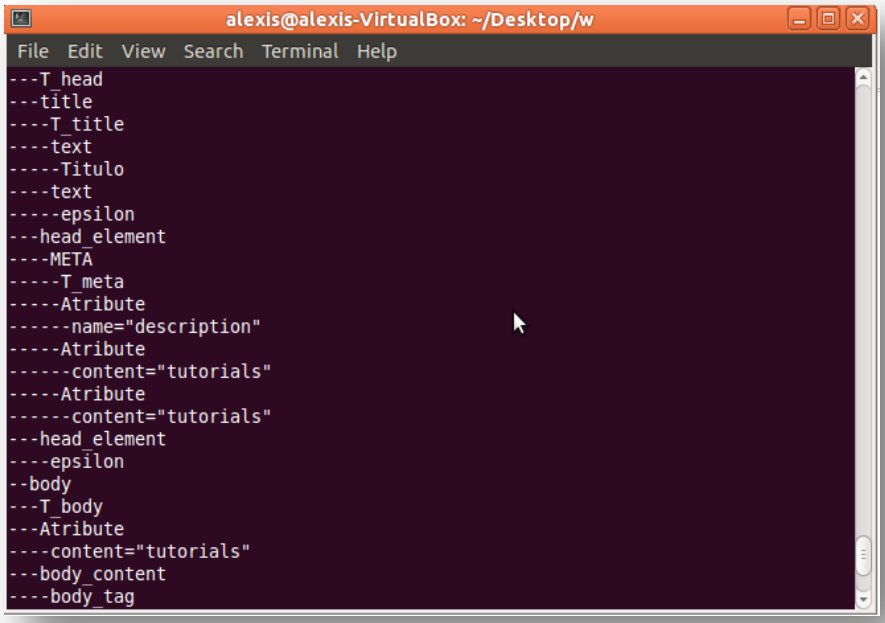


Ejemplo Árbol generado

Para mostrar gráficamente el árbol se tomó la idea de imprimirlo a manera como lo hace el comando **tree** en línea de comandos, el árbol mostrará todas las estructuras y subestructuras que lo conforman.



Comando Tree (Windows OS)



Árbol de análisis sintáctico representación grafica

Gramática utilizada

La gramática que se utilizó para este proyecto es creada por nosotros con la finalidad de ajustarnos a las especificaciones dadas y a las reglas gramaticales reales de XHTML, la misma es de tipo LALR típica del generador Bison. Se tomó como idea para la creación de la gramática una de las presentaciones del curso.

Ejemplo tomado de presentación del curso (ver referencias)

Bison

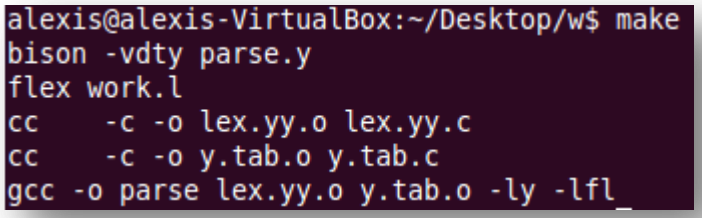
Escogimos Bison principalmente por su interacción con Flex el cual utilizamos para generar el analizador léxico en el proyecto anterior, además de que es posible encontrar mayor información y ejemplos en este generador que en su padre Yacc.

Análisis de Resultados

Gramática Correcta	**X**
Árbol de Análisis Sintáctico	X
Captura de Errores	X
Integración Flex-Bison	X

Manual del Usuario

El analizador cuenta con un archivo makefile para su compilación, el mismo puede ser encontrado en la carpeta del proyecto y basta con ejecutar desde un terminal el comando **MAKE.**



Luego de esto basta con ejecutar el analizador usando el comando. /**parser** con la posibilidad de analizar un archivo plano (./parse < xhtml.html).

El sistema comenzara su ejecución una vez ejecutados dichos pasos, y retornara el resultado dentro del mismo terminal.

Conclusión

- ✓ Bison como potente generador de analizadores sintácticos.
- ✓ XHTML es un lenguaje en el cual es fácil especificar gramáticas debido a sus robustas reglas de etiquetado.

Bibliografía

Scannig y Parsing en web browsers: http://www.tec-digital.itcr.ac.cr/dotlrn/classes/CA/IC5701/S-1-2013.CA.IC5701.2/file-storage/view/presentaciones/09_-_Scanning_y_parsing_en_web_browsers.pdf consultada el 27/05/2013

W3SCHOOL: http://www.w3schools.com/ consultada el 25/05/2013

Introducción Bison: http://ccia.ei.uvigo.es/docencia/PL/practicas0809/IntrodBison.html consultada el 25/05/2013

Bison: http://dinosaur.compilertools.net/bison/bison_5.html consultada el 25/05/2013

CStructs: http://www.cs.usfca.edu/~wolber/SoftwareDev/C/CStructs.htm consultada el 25/05/2013