

UNIVERSIDAD SAN CARLOS DE GUATEMALA -USAC
FACULTAD DE INGENIERÍA



Proyecto TytusX (Primera Fase)

Escuela de vacaciones Junio 2021

Organización de Lenguajes y compiladores 2

Catedrático: Ing. Luis Fernando Espino Barrios

Auxiliar: Haroldo Arias.

Alumnas :

Astrid Gabriela Martinez Castillo. 2017

Estrellita Guadalupe

Celia Esmeralda Vargas Lopez 20173093

Guatemala, 17 de junio 2021

ÍNDICE

INTRODUCCIÓN	3
OBJETIVOS	4
XML	5
XPATH	6
ANALIZADOR SINTACTICO ASCENDENTE	7
ANALIZADOR SINTACTICO DESCENDENTE	8
GRAMATICA XPATH	10
GRAMÁTICA XML	10

INTRODUCCIÓN

TytusX es un administrador de bases de datos documental de código abierto desarrollado bajo licencia MIT que utilizará lenguaje JavaScript para su construcción. Soportará archivos XML y manejará el lenguaje de consultas XPath.

OBJETIVOS

Objetivo General

- Presentar información técnica y puntual del proyecto TytusX para futuras implementaciones.
- Aplicar la fase de análisis y síntesis de la construcción de un compilador para realizar por un lado un intérprete y por otro lado un traductor a un código ejecutable utilizando herramientas de análisis ascendente.

Objetivos Específicos

- Generar un analizador léxico y sintáctico para construir un intérprete haciendo uso de atributos heredados y sintetizados.
- Usar tanto una gramática adecuada para el analizador ascendente y otra gramática adecuada para el analizador descendente

TytusX

XML

XML es el acrónimo de Extensible Markup Language, es decir, es un lenguaje de marcado que define un conjunto de reglas para la codificación de documentos.

El lenguaje de marcado es un conjunto de códigos que se pueden aplicar en el análisis de datos o la lectura de textos creados por computadoras o personas. El lenguaje XML proporciona una plataforma para definir elementos para crear un formato y generar un lenguaje personalizado.

Un archivo XML se divide en dos partes: prolog y body. La parte prolog consiste en metadatos administrativos, como declaración XML, instrucción de procesamiento opcional, declaración de tipo de documento y comentarios. La parte del body se compone de dos partes: estructural y de contenido (presente en los textos simples).

El diseño XML se centra en la simplicidad, la generalidad y la facilidad de uso y, por lo tanto, se utiliza para varios servicios web. Tanto es así que hay sistemas destinados a ayudar en la definición de lenguajes basados en XML, así como APIs que ayudan en el procesamiento de datos XML – que no deben confundirse con HTML.

XPATH

El lenguaje Xpath es el sistema que se utiliza para navegar y consultar los elementos y atributos contenidos en la estructura de un documento XML. Pero, para comprender el concepto de Xpath (XML Path Language), es importante explicar primero el significado de XML extensible Markup Language, cuya traducción es lenguaje de etiquetado extensible.

Para esto utiliza una sintaxis de criterios y funciones que le permiten realizar cálculos, comparaciones e identificar nodos. Podría decirse que XPath funciona de forma similar al editor de fórmulas en Excel, pues extrae información de un documento para hacer cálculos.

Sintaxis abreviada: más compacta y fácil de leer.

sintaxis completa: más larga pero con más opciones disponibles.

Las expresiones XPath se pueden dividir en pasos de búsqueda. Cada paso de búsqueda se puede a su vez dividir en tres partes:

- eje: indica el nodo o los nodos en los que se realiza la búsqueda.
- nodo de comprobación: especifica el nodo o los nodos seleccionados dentro del eje.
- predicado: permite restringir los nodos de comprobación.

ANÁLISIS SINTÁCTICO DESCENDENTE

En éste analizador las entradas son de izquierda a derecha, y construcciones de derivaciones por la izquierda de una sentencia o enunciado.

CARÁCTERÍSTICAS

- El análisis sintáctico descendente (ASD) intenta encontrar entre las producciones de la gramática la derivación por la izquierda del símbolo inicial para una cadena de entrada.
- Parte del axioma de la gramática.
- Procesa la entrada de izquierda a derecha.
- Escoge reglas gramaticales.
- Bueno primeramente para trabajar el análisis sintáctico descendente se debe realizar primeramente algunas operaciones para que la gramática sea LL1 las cuales son:
 - ELIMINAR AMBIGÜEDAD: Para eliminar la ambigüedad se debe reescribir la gramática.
 - ELIMINAR RECURSIVIDAD POR LA IZQUIERDA: Una gramática es recursiva por la izquierda si tiene un nodo Terminal a tal que existe una derivación $A \rightarrow A\alpha$ para alguna cadena . Es decir por simple observación podemos identificar.

ANÁLISIS SINTÁCTICO ASCENDENTE

El objetivo de un análisis ascendente consiste en construir el árbol sintáctico desde abajo hacia arriba, esto es, desde los tokens hacia el axioma inicial, lo cual disminuye el número de reglas mal aplicadas con respecto al caso descendente (si hablamos del caso con retroceso) o amplía el número de gramáticas susceptibles de ser analizadas (si hablamos del caso LL(1)).

- *Análisis ascendente con retroceso*

Al igual que ocurría con el caso descendente, este tipo de análisis intenta probar todas las posibles operaciones (reducciones y desplazamientos) mediante un método de fuerza bruta, hasta llegar al árbol sintáctico, o bien agotar todas las opciones, en cuyo caso la cadena se rechaza.

En el análisis con retroceso no se permiten las reglas ϵ , puesto que estas se podrán aplicar de forma indefinida.

- *Análisis Ascendente sin Retroceso*

El análisis ascendente sin retroceso busca una derivación derecha de la cadena de entrada de forma determinista.

Este se sustenta en su aplicación a las gramáticas LR(K).

La L viene de la lectura de la cadena de entrada de izquierda a derecha.

La R de producir un árbol de derivación derecho.

La k indica el número de símbolos que es necesario leer a la entrada para tomar la decisión de qué producción emplear.

Un parser del tipo shift-reduce puede verse como un autómata de pila determinista extendido que realiza el análisis de abajo hacia arriba.

Dada una cadena de entrada w , simula una derivación más a la derecha.

¿Cuál es su diferencia?

- ★ En el análisis sintáctico descendente: se construye el árbol sintáctico arriba hacia abajo y se utiliza más reglas.
- ★ En el análisis sintáctico ascendente: se construye el árbol sintáctico de abajo hacia arriba, lo cual disminuye el número de reglas mal aplicadas con respecto al caso descendente.

GRAMÁTICA XPATH Y XML

<https://drive.google.com/drive/folders/1xCnrACW58ycAawLPbcKiS1lyUU1kBj1t?usp=sharing>