

Actividad 3. Análisis Sintáctico

¿Qué es un análisis sintáctico aplicado a la valoración de un lenguaje formal?

Un análisis sintáctico, tanto en lenguajes formales como naturales, es el proceso de análisis de una sentencia, de acuerdo a las reglas gramaticales de dicho lenguaje. También es conocido como **parser**.

Éste análisis generalmente, es acompañado de un analizador léxico separado, el que le va alimentando los tokens, con estos tokens, busca crear una estructura de datos jerárquica, que puede ser un **árbol de parsing**, un **árbol de sintaxis abstracta** o alguna otra que represente la entrada de datos, al mismo tiempo que revisa que la sintaxis sea correcta.

Básicamente, existen dos formas de crear un análisis sintáctico:

Top-down parsing	Se inicia desde el axioma de la gramática y desarrolla por la izquierda el árbol de derivación sintáctica a medida que las sentencias son leídas.
LL	La cadena se inicia a leer por la izquierda y se hace la primera derivación por la izquierda
LL(k)	Se realiza un <i>preanálisis</i> de k caracteres, es decir se leen k caracteres con anticipación para poder predecir qué regla se debe de aplicar
Bottom-up parsing	Recorren el árbol de derivación sintáctica de abajo hacia arriba, es decir, desde las hojas hasta la raíz (axioma de la gramática)
LR	La cadena se lee de izquierda a derecha, pero las reglas de producción se aplican de forma inversa.

¿En qué consiste un análisis sintáctico y qué lo caracteriza?

En ésta fase, se reciben las secuencias de componentes léxicos identificadas en la fase anterior, y debido a que las combinaciones de estos componentes

Lo que busca el analizador sintáctico, es encontrar una forma de ir del símbolo de entrada a la cadena que se ingresó, mediante las reglas de derivación definidas en la gramática. Se puede ir de una cadena a otra de distintas formas, se podría derivar el primer no terminal a la izquierda, a la derecha, uno y uno, etc.

Con las derivaciones, se va construyendo el árbol de parsing, cada nodo interior está marcado con un símbolo no terminal y sus hijos, ordenados de izquierda a derecha, son la parte derecha de la regla de producción; y cada nodo hoja, está marcado con un símbolo terminal que leídos de izquierda a derecha, obtenemos una sentencia válida del lenguaje.

¿Qué casos de estudios son los importantes a considerar en el análisis sintáctico?

Los errores sintácticos son errores de estructura que se originan en el incumplimiento de alguna de las reglas de producción de la gramática y corresponden a los siguientes casos principalmente:

1. Cadenas de parentésis o corchetes no balanceados o ubicados incorrectamente
2. Cadenas no permitidas en estructuras sintácticas como consecuencia de la ausencia de operadores u operandos
3. Estructuras sintácticas alteradas por exceso o ausencia de delimitadores o separadores
4. Palabras reservadas mal formadas o usadas indebidamente
5. Bloques indebidamente delimitados por estar mal balanceados

El analizador debe de poder recuperarse de los errores generados, sin embargo, las estrategias son heurísticas y no tienen una validez general, las principales técnicas son:

1. **Recuperación en modo de pánico.** Cuando un error es detectado, avanza hasta completar la sentencia para volver a sincronizarse y reanudar el proceso.
2. **Recuperación por reemplazo.** Se realizan cambios en el código para intentar superar los errores detectados y luego informarlos.
3. **Recuperación por expansión de la gramática.** Se introducen reglas especiales en la gramática para poder superar ciertos errores, el objetivo es informar todos los errores una vez finalizado...

¿Qué procesos y problemas atiende un análisis sintáctico?

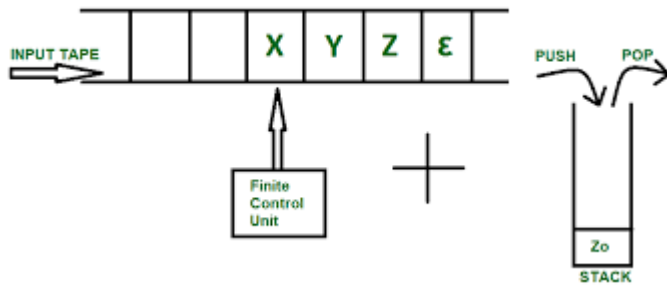
¿Cómo implementar utilizando un análisis sintáctico?

Catálogo de errores

Proponer los algoritmos y estructuras de datos necesarias para la implementación de un prototipo de analizador sintáctico

Para reconocer la sintaxis de una gramática libre de contexto, se requiere de un autómata de pila, que básicamente, es un autómata finito con una memoria de pila (estructura LIFO) que incrementa la posibilidad de memorizar total o parcialmente la cadena leída.

Su funcionamiento básico, se observa en la siguiente imagen



Podemos observar la cinta de lectura que cuenta con los tokens leídos, además cuenta con un control que va leyendo cada token y mantiene registro del estado actual; por último, hay una pila donde se van almacenando y recuperando tokens.

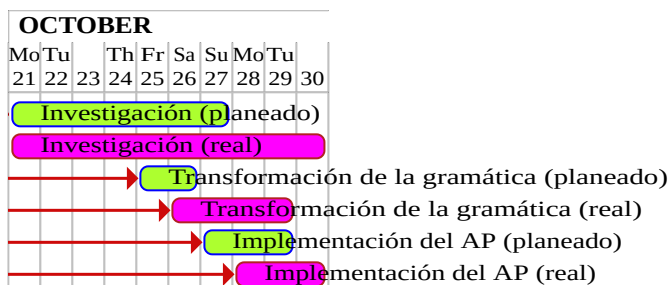
Diseño de una solución modelada en objetos, apoyada en diagramas UML que describan la arquitectura propuesta para el prototipo del analizador sintáctico

Modelado e implementación de un proceso de: lectura de código fuente => tokenización => categorización de los tokens => construcción y llenado de la tabla de símbolos => manejo y despliegue de errores => despliegue de la tabla de símbolos resultante => generación de una interface gráfica de operación => análisis sintáctico de los casos de estudio más importantes

Crear programas de mínimo 25 líneas cada uno, que ejemplifiquen los errores que en cada caso de estudio se propongan

Calendario de actividades planificadas vs. actividades realizadas

El calendario hasta el día 31 de octubre de 2019 es



Generación de una bitácora de incidencias

Día	Problema	Solución
23/10/2019	Inicié a desarrollar con base en el ejemplo del libro 'Compiladores: Principios, técnicas y herramientas' y descubrí que no es la mejor implementación	Cambiar la implementación a un autómata de pila
25/10/2019	No se puede obtener un autómata de pila de forma directa a partir de una gramática definida en BNF	Transformar la gramática a la Forma Normal de Chomsky y después a la Forma Normal de Greinbach
28/10/2019	Mi transformación a la Forma Normal de Greinbach no es correcta	Ir realizando la transformación poco a poco
30/10/2019	Las instrucciones if-else y do-while no se pueden anidar	No se ha solucionado

Código

El código completo, se encuentra en el [repositorio de Github](#) del proyecto