



Situación actual

Fco. Javier Bohórquez Ogalla

Índice

1. Visión general	3
2. Situación actual	3
3. Necesidades	4
4. Objetivos	4

1. Visión general

En esta sección se lleva a cabo un análisis de la situación actual en el área del aprendizaje de la teoría de compiladores y los lenguajes formales.

2. Situación actual

El estudio de los lenguajes formales es anterior a la concepción de las computadoras. Las matemáticas, la lógica y otras ciencias venían haciendo uso de los conceptos de los lenguajes formales para la solución de problemas.

Una computadora desde un punto de vista teórico es un autómata o máquina de estados, que es capaz de ejecutar una serie de instrucciones descritas en lenguaje máquina. Por tanto fueron los avances en la rama de la teoría de autómatas y los lenguajes formales, entre otros campos, los que permitieron la concepción de las primeras computadoras. Actualmente la mayoría de las ingenierías de la información se estudian los conceptos teóricos detrás de las computadoras y los lenguajes de programación.

Lo más común es que un estudiante de informática comience sus estudios en estos campos con los autómatas: los tipos que existen, cómo se definen y para qué se utilizan. Llegando a comprender conceptos como el de estado, alfabético, etc. Incluso estudiando definiciones de algunos tipos de lenguajes formales como los lenguajes regulares.

Posteriormente el alumno podría utilizar los conceptos aprendidos para estudiar distintos modelos de computación, los cuales son definidos formalmente y desde un punto de vista teórico. La máquina de Turing y el cálculo lambda son piezas esenciales en este punto del aprendizaje.

Una vez que se es poseedor del conocimiento base en la teoría de autómatas y los lenguajes formales, el alumno puede aplicar estos conocimientos para estudiar las estructuras, mecanismos y demás conceptos detrás de los lenguajes de programación. En este punto se estudian los intérpretes y compiladores, y los conceptos básicos que hay detrás de estos. Conceptos como el léxico, sintaxis y la semántica de los lenguajes de programación, las tablas de símbolos, o los distintos tipos de gramáticas.

Hasta aquí se habrá obtenido los conocimientos teóricos necesarios y el alumno podría comenzar a dar soluciones prácticas a problemas, aplicando así los conocimientos obtenidos. Así es común que se comience con el desarrollo de analizadores léxicos y sintácticos sencillos y concretos, para luego aplicarles una semántica. Ejercicios como el desarrollo de una calculadora suelen ser habituales. Además se estudian algunas de las herramientas que asisten al proceso del desarrollo de este tipo de tecnologías.

Después del proceso descrito al alumno se le ha brindado la oportunidad de profundizar en un campo con multitud de ramas, técnicas, metodologías y conceptos, que son fruto de años de estudio de expertos y apasionados. Podría profundizar en el proceso de compilación o traducción, en las distintas gramáticas, en técnicas de optimización o diseñar sus propias herramientas de traducción o interpretación de lenguajes formales.

Por otro lado, en la industria de la tecnología de la información se hace uso de lenguajes muy completos, con gran diversidad de mecanismos y bien consolidados, que son efecto de la evolución y las necesidades en el sector. En la mayoría de cursos académicos se estudian estas herramientas desde un punto de vista práctico y de uso.

La teoría de autómatas y lenguajes formales presenta la base para el estudio de los compiladores e intérpretes que son parte fundamental de la industria actual. A pesar de ello no existen herramientas divulgativas, colaborativas e interactivas que, a partir de los conocimientos básicos, ayude a comprender cómo se desarrollan los distintos mecanismos y herramientas presentes en la tecnología actual.

Los lenguajes de programación han evolucionado mucho desde que comenzaron a alejarse del lenguaje máquina. El alumnado actual trabaja sobre los conceptos que le ayudan a entender esta evolución, pero no dispone de herramientas o medios para ver cómo estos conceptos son trasladados a un producto real y presente en el día a día de un programador actual.

3. Necesidades

Dada la situación actual se precisa de una herramienta que ayude a comprender cómo se implementa y construye un intérprete para un lenguaje de programación. Es condición necesaria que este proceso quede correctamente documentado. La herramienta elaborada deberá ser accesible por cualquier interesado en el tema, que desee profundizar en la práctica del desarrollo de intérpretes y lenguajes de programación.

Se partirá de los conceptos básicos de la teoría de autómatas y los lenguajes formales, así como de la teoría de compiladores e intérpretes. Se asume pues que el usuario dispone de este conocimiento.

4. Objetivos

Se llegará a construir un lenguaje de programación completo con características presente en la tecnología actual. Este proceso quedará correctamente documentado y se pondrá a disposición pública. Las características que serán contempladas son:

- Distintos tipos de datos simples y compuestos.
- Expresiones lógicas y aritméticas.
- Expresiones y funciones sobre cadenas.
- Expresiones y funciones sobre vectores de datos.
- Expresiones y funciones sobre expresiones regulares.
- Operadores de conversión de tipos.
- Mecanismos de entrada/salida.
- Creación de procesos.
- Manipulación de ficheros.
- Funciones de fecha y tiempo.
- Definición y uso de variables de tipado dinámico.
- Ámbito de variables.
- Sentencias de control de flujo condicionales e iterativas.
- Definición y uso de funciones y procedimientos.
- Mecanismos y técnicas de la programación funcional.
- Definición y uso de clases de objetos.
- Mecanismos y técnicas de la programación orientada a objeto.
- Integración de módulos que extienden el lenguaje.

El interprete desarrollado podrá ser usado de una forma interactiva, permitiendo así la ejecución de instrucciones bajo demanda.

Toda la documentación generada a partir del proceso de desarrollo deberá ser estructurada y cumplimentada para formar una parte de una aplicación web que la haga accesible. La aplicación web además permitirá el uso online del intérprete.

El proyecto presentará una licencia de uso libre para que pueda ser usado abiertamente por la comunidad, a la vez de que se nutre de las contribuciones de la misma.