

Análisis estructuralista informal de la teoría de los lenguajes de programación

Mario Daniel Albarracín

Universidad Tecnológica Nacional, Facultad Regional Buenos Aires, Departamento de Ingeniería Química, Av. Medrano 951 (C1179AAQ), Ciudad Autónoma de Buenos Aires, Argentina

marioalbarracin58@gmail.com

Recibido el 5 de enero de 2023, aprobado el 31 de enero de 2023

Resumen

La teoría de los lenguajes de programación explica cómo se analizan, diseñan e implementan programas, de acuerdo al tipo de problema a resolver por una computadora. Además, el principal problema que presentan dichos lenguajes es la cantidad que tenemos actualmente en uso, además de las diferentes versiones que se han desarrollado. La intención a lo largo del trabajo ha consistido en mostrar cómo con el marco metateórico indicado, se puede elucidar claramente el carácter genuino de la teoría de los lenguajes de programación, lo que afirma la consideración de este marco metateórico.

PALABRAS CLAVE: LENGUAJES DE PROGRAMACIÓN - ESTRUCTURALISMO METATEÓRICO - MODELO - VÍNCULO - RED

Abstract

The theory of programming languages explains how they are analyzed, designed and implement programs, according to the type of problem to be solved by a computer. In addition, the main problem with these languages is the quantity that we currently have in use, in addition to the different versions that have been developed. The intention throughout the work has been to show how with the metatheoretical framework, we can clearly elucidate the genuine character of the theory of programming languages, which affirms the consideration of this framework metatheoretical.

KEYWORDS: PROGRAMMING LANGUAGES - METATHEORETICAL STRUTURALISM - MODELS - LINKS - NETWORK

Introducción

La concepción estructuralista de las teorías científicas tal como lo plantea Pablo Lorenzano, forma parte de las concepciones semánticas de la ciencia que constituyen el estructuralismo metateórico. Las concepciones semánticas identifican a las teorías con la clase de sus modelos o la secuencia de clases de modelos. La unidad más simple que identifica una teoría es el elemento teórico: $\langle K, I \rangle$.

K es el núcleo teórico, una estructura formada por los modelos potenciales (M_p), los modelos (M), los modelos parciales (M_{pp}), las condiciones de ligadura (C) y los vínculos interteóricos (L).

El modo de seleccionar la clase de modelos que caracteriza la identidad de una teoría es a través de un predicado conjuntista, definiendo un concepto de segundo orden en términos de la teoría de conjuntos.

M_p simboliza la clase total de entidades que satisfacen los axiomas impropios que caracterizan matemáticamente el aparato conceptual de la teoría, llamados modelos potenciales de la teoría. Los modelos actuales o modelos de la teoría M , simboliza a las entidades que satisfacen la totalidad de los axiomas introducidos, en otras palabras, que además de los axiomas impropios, satisfacen los axiomas propios o leyes fundamentales. La clase de modelos selecciona un subconjunto dentro del conjunto: $M_p: M \subseteq M_p$. Existen dos niveles conceptuales, el de los conceptos específicos de esa teoría T -teóricos y de los conceptos tomados de otra teorías T -no-teóricos. Esto se refleja en la diferencia de los conjuntos M_p y M_{pp} . Si al conjunto de los modelos potenciales M_p se le recortan los términos teóricos se obtienen los modelos parciales, que describen a través de conceptos no teóricos a la teoría en cuestión, los sistemas posibles a los que se puede aplicar dicha teoría. Su clase total se simboliza por M_{pp} . Si r es la función que recorta los componentes teóricos, tendremos: $M_{pp} := r(M_p)$.

Los modelos de la teoría están interconectados y forman una estructura global con conexiones cruzadas entre modelos denominadas condiciones de ligadura. Ciertas características se mantienen constantes a través de los distintos modelos, su clase total se simboliza por C . La clase de condiciones de ligadura selecciona un subconjunto dentro del conjunto potencial de M_p y excluye a los miembros que no pertenecen a ese subconjunto: $C \subseteq \text{Pot}(M_p)$.

Distintas teorías relacionadas entre sí constituyen la clase total de dichas relaciones interteóricas llamadas vínculos, simbolizada L , donde la clase vínculo selecciona un subconjunto dentro del conjunto dado por $M_p: L \subseteq M_p$. El núcleo K constituye la parte formal de la teoría donde: $K = \langle M_p, M, M_{pp}, C, L \rangle$.

I es el conjunto de las aplicaciones intencionales de la teoría, el sistema empírico donde queremos aplicar la teoría, de manera que $I \subseteq M_{pp}$ es todo lo que se puede decir sobre el conjunto I de aplicaciones propuestas. El campo I es un conjunto abierto y no puede definirse a partir de la introducción de condiciones necesarias y suficientes para su pertenencia y su extensión no puede ser dada de una vez, para siempre. Es un concepto pragmático y diacrónico.

El núcleo K de elementos teóricos T junto con sus leyes fundamentales se aplica al campo de aplicaciones intencionales I y puede ser tratado por medio del núcleo K o subsumidas bajo el núcleo K del elemento teórico T . Esto se explica mediante una aserción empírica de la teoría. Ciertos sistemas empíricos descriptos T -no-teóricos se comportan según las restricciones legales: leyes, condiciones de ligadura y vínculos interteóricos determinan el nivel T -no-teórico. De esta manera, todo sistema propuesto dado puede ser, agregando un conjunto de componentes T -teóricos a los T -no-teórica de $K(T)$, extendido a, o incrustado

en, un modelo de $M(T)$, o extendido a un $Mp(T)$ e incrustado en un modelo de $M(T)$, que cumpla con los vínculos interteóricos $L(T)$ y las condiciones de ligadura $C(T)$.

Es común que las teorías se identifiquen con más de un elemento teórico. De esta forma una teoría se puede pensar como un conjunto de elementos teóricos producto de la existencia de diferentes leyes con distintos niveles de generalidad en la misma teoría conformando redes teóricas. Se denominan leyes fundamentales a aquellas leyes que son válidas para todas las aplicaciones pretendidas y leyes especiales a las que son válidas para un subconjunto de las mismas. Se denomina elemento teórico básico al elemento teórico que contiene la ley fundamental o las leyes fundamentales de la teoría.

Por otra parte, se denominan elementos teóricos especiales o especializaciones de la teoría a los que tienen leyes especiales. Las especializaciones se obtienen introduciendo sucesivas restricciones a la ley fundamental especificadas por sus componentes, aumentando su contenido empírico. Se define como red teórica a un conjunto de elementos teóricos relacionados entre sí. Podemos decir entonces que una red teórica N es un conjunto de elementos teóricos interconectados a través de una relación de especialización. Partiendo de un elemento teórico básico, se van concretando progresivamente en direcciones diversas cada vez más restrictivas y específica, las ramas de la red teórica, denominadas redes teóricas arbóreas o árboles.

Reconstrucción informal de la teoría de los lenguajes de programación

A partir del conjunto de instrucciones dadas a la computadora, esta procesa los datos de los cuales se obtendrá la información buscada. Se deben dar a la máquina las instrucciones que conforman el programa. Para que se pueda diseñar una solución a un problema dado que pueda ser realizada por una computadora, debemos distinguir el análisis del problema y la programación.

El análisis del problema que permite entenderlo y poder resolverlo, en el entorno de la máquina, se divide en tres partes: identificar los datos, determinar los resultados y explicitar las operaciones que vinculen a los datos con los resultados.

Son pocas las operaciones que puede realizar una computadora:

- Operaciones aritméticas sencillas: suma, resta, multiplicación y división.
- Operaciones lógicas sencillas: unión, intersección y complemento.
- Operaciones relacionales: mayor, menor, igual y distinto.
- Almacenar y recuperar información.

En el terreno de las ciencias de la computación, un programa es un conjunto de instrucciones. Programar es explicitar las instrucciones en la computadora. La idea es poder escribir instrucciones a partir de operaciones sencillas de tal forma que dando un orden lógico a dichas instrucciones, generemos un programa que le permita a la computadora, resolver la tarea que le fue encomendada. Es decir que para llevar adelante una acción, se debe ejecutar una instrucción que implica una operación por parte de la máquina. La computadora, por lo tanto, funcionará de acuerdo a quién le escriba las instrucciones para programarla. Pienso una acción y escribo una instrucción para que la máquina realice una operación de cálculo, de almacenamiento o de recuperación de la información.

Un programa consiste en una secuencia de instrucciones que ha de procesar la computadora con el objetivo de obtener resultados. Los datos a utilizar en las operaciones son entidades, la materia prima de la información. Los datos se representan por medio de símbolos.

Un programa se estructura en tres partes:

- **Entrada de datos:** está formada por todas las instrucciones que toman los datos objeto del programa, desde un dispositivo externo (unidad de entrada) depositándolos en la memoria principal de la computadora, incluyendo la depuración o validación de los mismos.
- **Proceso:** Conjunto de instrucciones que resuelven el problema a partir de los datos que han sido introducidos, dejando los resultados en la memoria principal. El dispositivo físico encargado de llevar a cabo esta tarea es la unidad central de proceso.
- **Salida de resultados:** La constituyen las instrucciones que hacen que los datos resultantes del proceso, sean proporcionados al exterior por medio de algún dispositivo (unidad de salida).

Estos tres componentes de todo programa no están separados sino que las instrucciones pertenecientes a cada uno de ellos están mezcladas entre sí, pues es común realizar operaciones de entrada después de iniciar un proceso y proporcionar algunos resultados antes de terminado los mismos. La manera que hay que indicarle a la computadora lo que tiene que hacer un programa, se compone de variables, constantes y operadores. Las variables y las constantes pueden ser numéricas o alfanuméricas. Los operadores pueden ser aritméticos, alfanuméricos, lógicos y relacionales.

En el diseño de un programa se utilizan en forma combinada los métodos de programación modular y programación estructurada. La programación modular se basa en descomponer una determinada acción compleja en función de un número de acciones más simples, capaces de ser ejecutadas por una computadora y que constituyen sus instrucciones. Para la descomposición se utiliza el diseño descendente, que es el proceso mediante el cual un problema se descompone en una serie de niveles o pasos sucesivos de refinamiento. La metodología descendente consiste en efectuar una relación entre las sucesivas etapas de la estructuración, de modo que se relacionen unas con otras mediante entradas y salidas de datos. Es decir, se descompone el problema en etapas y estructuras jerárquicas, de modo que se puede considerar cada estructura desde dos puntos de vista: desde el exterior, que hace y desde el interior, como lo hace.

El conjunto de instrucciones que constituyen un programa, se agrupa a su vez en subconjuntos denominados estructuras. El Teorema de las estructuras muestra cómo cualquier programa, se puede escribir utilizando tres estructuras de control: secuencial, decisión y repetición. Un programa se define como propio si cumple las siguientes características: posee un solo punto de entrada y uno solo de salida o fin para control del programa, existen caminos desde la entrada hasta la salida que pueden seguir y que pasan por todas partes del programa, todas las instrucciones son ejecutables y no existen lazos o bucles infinitos.

La teoría de los lenguajes de programación explica cómo se analizan, diseñan e implementan programas, de acuerdo al tipo de problema a resolver por una computadora. Se trata de proveer a la máquina, las instrucciones necesarias para que pueda llevar adelante los trabajos planteados. Se denomina programa a un grupo ordenado y finito de instrucciones que se dan a una computadora para realizar un proceso determinado. Una instrucción consiste en una secuencia de datos de entrada que ha de procesar la computadora con el objetivo de obtener unos resultados o datos de salida.

Modelos Potenciales

M_p (TLP) simboliza la clase total de entidades que satisfacen los axiomas impropios que caracterizan matemáticamente el aparato conceptual de la teoría, llamados modelos potenciales de la teoría. La TLP explica la forma como se escribe una secuencia de instrucciones que ha de procesar la computadora con el objetivo de obtener unos resultados o datos de salida a partir de unos datos de entrada.

Los datos se representan por medio de símbolos:

- Símbolos numéricos: números enteros y números reales.
- Símbolos alfanuméricos: letras, números y caracteres especiales.
- Símbolos booleanos: verdadero y falso.

Los datos se cargan como constantes o como variables:

- Constantes: números enteros, números reales y símbolos alfanuméricos.
- Variables: números enteros, números reales y símbolos alfanuméricos.

Las constantes y las variables se combinan con operadores y caracteres especiales:

- Operadores aritméticos: + (suma), - (resta), * (multiplicación), / (división) y ^ (potenciación).
- Operadores alfanuméricos: # (concatenación).
- Operadores relacionales: > (mayor), < (menor), == (igual), <> (distinto), >= (mayor o igual) y <= (menor o igual).
- Operadores lógicos: || (or), && (and) y != (not).
- Caracteres especiales: , (coma), “...” (comillas) y (...) (paréntesis).

A partir de operaciones sencillas realizadas con constantes, variables, operadores y caracteres especiales, construimos instrucciones para las cuales nos valdremos de letras mayúsculas para las palabras reservadas del pseudocódigo utilizado para representar dichas instrucciones:

- La instrucción de entrada **INGRESAR** es un conjunto de constantes, variables y caracteres alfanuméricos.
- La instrucción de salida **IMPRIMIR** es un conjunto de constantes, variables y caracteres alfanuméricos.
- La instrucción de asignación, variable = expresión, es una función de variables, constantes y operadores aritméticos.
- La instrucción de decisión **SI ENTONCES SINO** es una función de variables, constantes, operadores aritméticos, operadores lógicos y operadores relacionales.
- La instrucción de decisión **SELECCIONAR CASO** es una función de variables, constantes, operadores aritméticos, operadores lógicos y operadores relacionales.
- La instrucción de repetición **HACER MIENTRAS REPETIR** es una función de variables, constantes, operadores aritméticos, operadores lógicos y operadores relacionales.
- La instrucción de repetición **PARA PRÓXIMO** es una función de variables, constantes, operadores aritméticos, operadores lógicos y operadores relacionales.
- La instrucción de arranque **COMIENZO** es una orden de comenzar.
- La instrucción de detención **FIN** es una orden de finalizar.

Con el conjunto de instrucciones generamos un programa que le permita a la computadora, resolver la tarea que le fue encomendada.

Conceptos T-Teóricos: Modelos

Un concepto es T-teórico, si es propio de la teoría, su determinación depende siempre de T. La instrucción es un procedimiento de determinación que depende de la TLP pues presupone la aplicabilidad de la teoría. Definimos los modelos de los que tiene sentido preguntarnos si cumplen con los axiomas propios de la teoría.

Las estructuras que además de cumplir con los axiomas impropios, cumplen con los propios o leyes de la teoría, se denominan modelos M(TLP): la instrucción de entrada **INGRESAR**, la instrucción de salida **IMPRIMIR**, la instrucción de asignación, la instruc-

ción de decisión SI ENTONCES SINO, la instrucción de decisión SELECCIONAR CASO, la instrucción de repetición HACER MIENTRAS REPETIR, la instrucción de repetición PARA PRÓXIMO, la instrucción de arranque COMIENZO y la instrucción de detención FIN.

Conceptos T-No Teóricos: Modelos Parciales

Un concepto es T-no teórico, si es anterior a la teoría de los lenguajes de programación, tiene procedimientos de determinación independientes de la teoría, dado que es posible determinarlo sin suponer la aplicación de la teoría.

Si al conjunto de modelos potenciales M_p , se les recortan los términos teóricos, se obtienen los modelos parciales $M_{pp}(TLP)$, que describen a la TLP: símbolos numéricos, símbolos alfanuméricos, símbolos booleanos, constantes, variables, operadores aritméticos, operadores alfanuméricos, operadores relacionales, operadores lógicos, caracteres especiales.

Condiciones de ligadura

El conjunto de instrucciones que constituyen el programa, no es el único que impone condiciones adicionales efectivas al modelo potencial de la teoría de los lenguajes de programación. Si se considera el modelo suelto, si, pero considerando varios modelos sueltos, no. No se puede expresar con axiomas normales que involucran modelos sueltos.

Las condiciones que definen las ligaduras **C(TLP)** son las estructuras. La idea es determinar un conjunto de modelos. Definen el tipo de programación como estructurada; puedo visualizar un programa como un conjunto de estructuras compuestas por instrucciones:

- Estructura secuencial (SEC): las instrucciones se ejecutan en el orden que se han codificado.

Sintaxis:

```

Instrucción 1
Instrucción 2
.....
Instrucción N
    
```

- Estructura de decisión (DEC): controla el flujo de ejecución de la instrucción al seleccionar que instrucción o grupo de instrucciones deben ejecutarse. Esto depende de si se cumple o no, una condición evaluada. La estructura evalúa una condición, si la condición es verdadera, se ejecuta un grupo de instrucciones, sino se ejecuta otro grupo de instrucciones. Existen distintos tipos de instrucciones de decisión que permiten armar una estructura de decisión: simple, doble, en bloques y múltiple.

Sintaxis de decisión simple:

SI (condición) ENTONCES instrucción

Si la condición es verdadera se ejecuta la instrucción.

Sintaxis de decisión doble:

SI (condición) ENTONCES instrucción 1 SINO instrucción 2

Si la condición es verdadera se ejecuta la Instrucción 1, sino se ejecuta la Instrucción 2.

Sintaxis de decisión en bloque:

SI (condición) ENTONCES

Instrucción 1

SINO

Instrucción 2

FIN SI

Sintaxis de decisión múltiple:

SELECCIONAR CASO variable

CASO condición 1

Instrucción 1

CASO condición 2

Instrucción 2

.....

OTRO CASO

Instrucción N

FIN SELECCIONAR

- Estructura de repetición (REP): este tipo de estructura nos da la posibilidad de repetir un conjunto de instrucciones un número determinado de veces. De acuerdo a la forma de control de la cantidad de repeticiones podemos diferenciar dos grupos de instrucciones: controladas por condición y controladas por contador. En el primer caso un grupo de instrucciones se ejecutan una determinada cantidad de veces mientras se cumpla una determinada condición hasta que deje de cumplirse. Entonces sale del ciclo y continúa ejecutando las instrucciones que están a continuación.

Sintaxis:

HACER MIENTRAS (condición)

Instrucción 1

Instrucción 2

Instrucción 3

.....

Instrucción N

REPETIR

En el segundo caso se conocen la cantidad de veces que se tiene que repetir la ejecución del bloque de instrucciones que conforma la estructura de repetición.

Sintaxis:

PARA índice = valor inicial A valor final PASO incremento

Instrucción 1

Instrucción 2

Instrucción 3

.....

Instrucción N

PRÓXIMO

El bloque de instrucciones comienza a repetirse de acuerdo al contador que está definido por el valor inicial de arranque que toma la variable índice, el valor final de parada y un “paso” que indica el incremento que tiene en cada vuelta. Si el paso es uno, su escritura es opcional.

Vínculos interteóricos de la Teoría de los Lenguajes de Programación L (TLP)

Los elementos teóricos no se encuentran aislados del resto de las teorías. Las teorías que se asignan en las instrucciones van a constituir los vínculos interteóricos L(TDP).

Los vínculos interteóricos que conectan la TLP con otras teorías a través de términos TLP no teóricos provenientes de otras teorías son:

Modelo de la teoría de estructura de datos, M(EDD)

Un conjunto de datos homogéneo que se tratan como una sola unidad se denomina estructura de datos.

Si una estructura de datos reside en la memoria puede ser estática pues tiene un número fijo de elementos que queda determinado desde la declaración de la estructura en el comienzo del programa. La estructura de datos estática es la tabla. Una tabla consiste en un número fijo, finito y ordenado de elementos del mismo tipo y bajo un nombre común para todos.

Según su dimensión se clasifican en:

- Vector: conjunto de variables ordenadas bajo un mismo nombre a partir de un índice.
- Matriz: conjunto de variables ordenadas bajo un mismo nombre a partir de dos índices.
- Multidimensional: conjunto de variables ordenadas bajo un mismo nombre a partir de tres o más índices.

Si una estructura de datos que reside en la memoria es dinámica. Se clasifican en:

- Listas: es una estructura de datos interna consistente en una secuencia lógica de elementos del mismo tipo.
- Pilas: es una lista en la que las inserciones, consultas y supresiones pueden hacerse únicamente por un extremo denominado cima. Se las denomina lista LIFO, último en entrar primero en salir (*last input first output*).
- Colas: es una lista en la que las inserciones se realizan por uno de sus extremos, llamado FINAL, y las supresiones por el otro llamado Frente. Se denomina lista FIFO, primero en entrar primero en salir (*first in first out*).

Las estructuras de datos no lineales pueden clasificarse como:

- Árboles: es un conjunto de nodos dónde existe un nodo raíz y el resto sub árboles del raíz.
- Grafos: es un conjunto de puntos(nodos) y líneas (arcos), cada una de las cuales une un punto a otro.

Modelo de la teoría de sistemas operativos, M(ARC)

Un sistema operativo es un conjunto de programas y funciones que controlan el funcionamiento del hardware ocultando sus detalles, ofreciendo al usuario una vía sencilla y flexible de acceso.

Una computadora es una máquina que posee un conjunto de elementos denominados recursos, que deben ser racionalmente distribuidos y utilizados para obtener de ellos el mejor rendimiento. Estos recursos son:

- El procesador: el lugar donde se ejecutan las instrucciones y por ello deben controlarse los programas que se ejecutan y su secuencia.
- La memoria principal: todo programa que se ejecute en la computadora y todo dato que se desee procesar, debe residir en la memoria principal y debe regularse su uso y ocupación.
- La entrada/salida: todo programa necesita realizar operaciones de entrada/salida sobre sus unidades periféricas para el control y direccionamiento de las mismas.
- La información: los datos, sus tipos, tamaños y métodos de representación tienen que es-

tar perfectamente controlados para evitar operaciones erróneas o falsas interpretaciones.

Modelo de la teoría de archivos, M(ARC)

Un archivo es una estructura de datos que reside en memoria secundaria, consistente en un conjunto de información estructurada en unidades de acceso denominadas registros, todos del mismo tipo y en números indeterminados. Un registro es cada uno de los componentes del archivo, conteniendo el conjunto de informaciones que se acceden y se tratan de manera unitaria. Está constituido por uno o más elementos denominados campos, que pueden ser de diferentes tipos y que a su vez pueden estar compuestos por sub campos. Un registro puede tener un campo clave, cuyo valor sirve para identificar de forma única el registro y por tanto dicho valor no puede aparecer repetido en otro registro diferente.

Puede suceder que un archivo no tenga campo clave en su registro o que tenga varios, denominándose a la principal clave primaria y a las demás secundarias.

Si un archivo contiene la información de un conjunto de individuos u objetos, sus registros contienen la información de cada uno de ellos y los campos los diferentes datos que los componen.

Modelo de la teoría de base de datos, M(BDD)

Una base de datos es un conjunto integrado de datos interrelacionados, junto con una serie de aplicaciones para su manejo, accesibles simultáneamente por diferentes usuarios. El modelo de base de datos hace referencia a la estructura que se utiliza para expresar las relaciones existentes entre las diferentes unidades de datos que las constituyen. Entre las posibilidades útiles para ello, son tres los modelos que se han afianzado y están actualmente en uso:

- Modelo jerárquico: utiliza la estructura de árbol para establecer relaciones del tipo 1:n (uno a muchos). Una base de datos de este tipo consistirá en uno o varios árboles que expresarán las distintas agrupaciones entre los datos. Cada árbol tiene un nodo distinto denominado raíz y, a partir de él, se establecen relaciones de sucesión, denominándose nodos hijos a los descendientes de otro. Mientras que un nodo padre sólo puede tener una ocurrencia, los nodos hijos pueden tener varios.
- Modelo en red: utiliza la estructura de red o grafo, que permite definir entre todos los nodos relaciones n:n (muchos a muchos). Mediante el establecimiento de apuntadores entre nodos se puede relacionar cualquier unidad de datos con cualquiera de los otros.
- Modelo relacional: las relaciones entre las unidades de datos o entidades se expresan mediante tablas de dos dimensiones. Una base de datos de este tipo estará formada por varias de estas tablas en las que una fila contiene una ocurrencia de valores interrelacionados y una columna contiene los diferentes valores posibles de cada entidad.

Modelo de la teoría sistemas expertos, M(INA)

Un sistema experto soluciona problemas complejos, simulando el proceso de razonamiento humano mediante la aplicación específica de conocimientos. Se emplea para ejecutar tareas que en el pasado llevaban a cabo expertos. A través de la aplicación de las técnicas de inteligencia artificial, los sistemas expertos captan el conocimiento básico que permite desempeñarse como un experto frente a problemas complejos. La característica de los sistemas expertos, que los diferencia de las tradicionales de las computación, es la capacidad que poseen para enfrentar problemas complejos que constituyen el mundo real, por medio de la aplicación del proceso que reflejan el razonamiento y la percepción humana.

Un sistema experto está compuesto por:

- Amplio conocimiento específico a partir del campo de interés.
- Aplicación de técnicas de búsqueda.
- Soporte para el análisis heurístico.
- Habilidad para inferir nuevos conocimientos a partir de conocimientos existentes.
- Procesamiento de símbolos.
- Capacidad para explicar su propio razonamiento.

Un usuario de un sistema experto puede estar trabajando en cuatro modos distintos:

- Verificador: intenta comprobar la validez del sistema.
- Tutor: modifica el conocimiento presente en el sistema.
- Alumno: se capacita en el área específica mediante el sistema.
- Cliente: aplica el sistema a tareas reales.

Expresamos el efecto constrictivo conjunto de todos los vínculos como vínculo global GL, a través de la intersección conjunta de todos ellos. Esto nos permite caracterizar el núcleo teórico de la teoría de programación TDP de la siguiente manera:

$$K(TLP) = \langle Mp(TLP), M(TLP), Mpp(TLP), GC(TLP), GL(TLP) \rangle$$

Aplicaciones propuestas o intencionales

El dominio de la aplicación pretendida o intencional $I(TLP)$, es el conjunto de los sistemas empíricos a los que se quiere aplicar la teoría.

Existen un conjunto I de algoritmos clásicos que constituyen las aplicaciones pretendidas y que conforman la base empírica de la teoría. Contienen elementos pragmáticos que no se pueden eliminar pues su determinación es intencional y paradigmática. Es intencional porque el conjunto I de algoritmos, es objeto intencional de los usuarios de la teoría. Es paradigmática porque el conjunto I de algoritmos, no se caracteriza presentando todos los sistemas de información que son aplicaciones pretendidas y sus ejemplos paradigmáticamente.

Conjunto de aplicaciones paradigmáticas propuestas son los algoritmos clásicos:

- Carga: carga de datos.
- Acumulador: acumula valores constantes y variables.
- Cálculo: realiza cálculos matemáticos.
- Búsqueda: busca y recupera información.
- Ordenamiento: ordena en forma ascendente o descendente un vector.
- Intercalación: mezcla dos vectores ordenadas produciendo un nuevo vector ordenado.
- Impresión: imprime variables y constantes, numéricas y alfanuméricas.
- Almacenamiento: almacena variables y constantes, numéricas y alfanuméricas.

Un elemento teórico T se identifica por el par formado por el núcleo K , la parte formal y el dominio de aplicación I , la parte aplicativa.

$$T(TLP) := \langle K(TLP), I(TLP) \rangle$$

El conjunto de las aplicaciones intencionales más el conjunto de las aplicaciones paradigmáticas nos permite determinar el elemento teórico básico de TLP.

Especialización: Red teórica

Los programas complejos se dividen en un programa principal y un conjunto de subprogramas. Los subprogramas se dividen a su vez en funciones y procedimientos, la diferencia entre ellos es que un procedimiento es un subprograma que ejecuta una serie de acciones que están relacionadas entre sí, puede devolver múltiples valores o ninguno,

escribir en pantalla o leer datos, mientras que una función es un subprograma que recibe los valores de uno o varios argumentos y devuelve como mínimo un valor.

La teoría de los lenguajes de programación utiliza funciones FUN y procedimientos PRO para conformar un programa P. La primera rama de especialización especifica los componentes TLP-teóricos de un subprograma de funciones: TLPF.

La segunda rama de especialización especifica los componentes TDP-teórico de un subprograma de procedimientos: TLPP.

A continuación se representa la red teórica de la teoría de programación TLP a través de un esquema en el cual, pueden observarse las relaciones de ambas ramas de especialización entre los elementos teóricos de los subprogramas de funciones TLPF y procedimientos TLPP:

$$TLP = TLPF + TLPP$$

Conclusiones

Los trabajos de Corrado Böhm y Giuseppe Jacopini en la utilización de diagramas de flujo, Edsger Dijkstra en la derivación de programas y de refinamiento sucesivo de Niklaus Wirth cobran un sentido mucho más claro. Lo que están haciendo Böhm, Jacopini, Dijkstra y Wirth en estos casos es aplicar la teoría de los lenguajes de programación a partir de sus leyes especiales. Sus argumentaciones se vuelven efectivas retóricamente y sólidas.

Se ha tratado de brindar una interpretación de la teoría de los lenguajes de programación a partir de una reconstrucción informal y para que mis puntos de vistas se vuelvan más convincentes, tendría que formular la estructura formal de la TLP. Sin embargo creo que una interpretación de la ley fundamental de la TLP que permite explicar su carácter genuino como ley fundamental, elimina las razones por las cuales algunos pueden considerar que carece de ese carácter genuino. La capacidad de un enfoque basado en el estructuralismo metateórico para explicar el error de los enfoques opuestos, es un argumento importante a favor.

La razón por la cual no se realizó una reconstrucción formal, tarea que queda pendiente, está dada porque se considera que primero debe existir una reconstrucción informal para luego pasar a realizar la reconstrucción formal que es verdaderamente compleja y quitaría claridad a este trabajo.

Aunque no se brindó una reconstrucción formal de la teoría de los lenguajes de programación, este planteo depende de manera absoluta del enfoque desde el cual se analizó: el estructuralismo metateórico. El contar con marcos conceptuales adecuados es fundamental para pensar cuestiones particulares de la ciencia dado que estos marcos son necesarios si se nutren de lo que los filósofos de la ciencia describen al analizar prácticas científicas particulares.

Alan Turing, científico inglés que presentó una máquina teórica de capacidad infinita que trabajaba a partir de un conjunto de instrucciones lógicas, definiendo de esta manera el concepto moderno de algoritmo. Plantea también que existen problemas sin solución algorítmica, constituyendo de esta forma, el comienzo de la teoría de la computación, explicando en un lenguaje matemático y de manera precisa, cómo puede un sistema automático, trabajando sobre la base de reglas muy simples, resolver toda clase de operaciones matemáticas escritas en un lenguaje formal.

La máquina de Turing de carácter virtual demostraba a partir de su teoría de computación, que se podía construir determinado tipo de máquina computadora para re-

presentar un proceso a partir de su descripción. El objetivo de Turing era diseñar una máquina que interpretara los procesos de cómputo. Las máquinas de Turing son útiles como herramientas para investigar las limitaciones de las computadoras actuales y de los procesos algorítmicos.

A partir de la arquitectura física de las computadoras, John Von Neumann desarrolló la idea de algoritmo interno, una alternativa hacia una arquitectura más flexible que permitió intercambiar algoritmos y que trabajara en sistema binario. Todo un nuevo sistema lógico de computación basado en las ideas de Turing.

Referencias

- ALBARRACÍN, M.; ALCALDE, E.; GARCÍA, M., (1996). "Introducción a la Informática", Chile, Mc Graw Hill, pp.27-52.
- BOHM, C. y JACOPINI, G., (1966). "Flow Diagrams, Turing Machines and Languages with Only Two Formation Rules". Communications of the ACM, Vol.9, No 5.
- DIJKSTRA, E., (1972). "Go To Statement Considered Harmful". Communications of the ACM, Vol.11, No3, pp.147-148.
- LORENZANO, P., (2004). "Filosofía de la ciencia", Bernal: Universidad Virtual de Quilmes.
- TURING, A. M., (1936). "On computable numbers, with an application to the Entscheidungs problem", Proceedings of the London Mathematical Society, Series 2, 42, pp 230 – 265.
- TURING, A.M., (1950). "Computing machinery and intelligence". Mind, 59, 433-460.
- VON NEUMANN, J., (1946). "The Moore School Lectures", MIT press, Vol. IV, conf.40.
- WIRTH, N., (1980). "Algoritmos + Estructuras de Datos = Programas". Ediciones del Castillo.