Paradigmas de programación Lógico y Funcional

Saul Ivan Rivas Vega
Tarea 1
Programación Avanzada

15 de agosto de 2019

1. Introducción

Una definición para el paradigma de programación es la interrelación entre algoritmos y datos, y que esta ha estado en constante evolución [1] y esto se ha visto reflejado en la creación de nuevos paradigmas encontrando nuevas formas de ver esa interrelación dependiendo del caso de uso.

Es también una forma o estilo de programar, algunos lenguajes tendrán entonces una mayor facilidad de escribir en ciertos paradigmas que otros. [2]

En este trabajo se hablará de los paradigmas lógico y funcional mencionando sus principales características, ventajas y desventajas así como los mas conocidos lenguajes que los representan.

2. Desarrollo

2.1. Paradigma Funcional

2.1.1. Características

En el paradigma funcional el programa principal es una función que recibe los parámetros de entrada como argumento y devuelve el resultado como

salida, este programa principal esta compuesto de funciones y estas a su vez están compuestas de mas funciones quedando en el mas bajo nivel de definición los valores primitivos del lenguaje. [3].

Estas funciones también deben ser puras, esto quiere decir que las funciones solo dependen de sus argumentos, sin hacer referencia o uso de variables en el entorno global, ademas de que no debe haber efectos secundarios, para preservar la inmutabilidad de las variables y el cambio de estado que influya en la evaluación de la función. [4]

Podemos entonces hablar en el caso práctico que una función en un programa para que pueda ser llamada pura debe contar con variables locales que almacenen el resultado de alguna operación realizada a un elemento de entrada, por ejemplo, una función que calcule el resultado de alguna operación aritmética a un número esta función debe cumplir con que:

- No debe hacer uso de valores fuera de la definición de la función.
- No debe modificar la variable que recibe como entrada.

Lo cual dejaría como opción crear una variable que contenga el resultado de la operación sobre el número, como una multiplicación por el valor de una constante definida en la función u otro valor que recibiera como argumento, y devolver dicha variable.

2.1.2. Metodologías

Mencionaremos algunos de los métodos que se usan en la programación funcional como es el caso de las funciones de orden superior, las cuales reciben otras funciones como parte de sus argumentos o devuelven como resultado otra función. [5].

filter Se recibe una colección de datos y una función de filtrado como entrada y devuelve una colección de datos que contiene a los elementos que cumplan con la función de filtrado.

map Se recibe una colección de datos y una función de modificación como entrada y devuelve una colección de datos que contiene el resultado de aplicar la función de modificación a cada elemento.

reduce Se recibe una colección de elementos y una función de reducción como entrada y devuelve un solo dato que es el resultado de combinar los elementos usando la función de reducción, por ejemplo la suma de todos los elementos.

2.1.3. Lenguajes

Algunos de los mas populares lenguajes usados para el paradigma funcional son **Haskell**, **Scala**, **F#** e incluso con base en las propiedades que se requieren se pueden considerar lenguajes como **Python**, **Javascript** y **C++11**. Ahora si consideramos las características definidas y las capacidades que han mejorado en los lenguajes de programación podemos entonces registrar que tanto cumplen con el paradigma funcional los lenguajes llegando a representaciones como en 1.

How Functional Is My Language?	Agda	C	C++11	C++17	Clojure	Common LISP	F#	Haskell	Java 8	JavaScript	Perl	Python	Racket	Rust	Scala	Scheme	SML & OCaml
pure	Y	N	N	N	N	N	N	Y	N	N	N	N	N	N	N	N	N
proper tail calls	N/A	N	N	N	M	M	Y	Y	N	N	N	N	Y	N	M	Y	Y
higher-order	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
function type	Y	M	Y	Y	N/A	N/A	Y	Y	M	N/A	N/A	N/A	N/A	Y	Y	N/A	Y
lambda	Y	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
statement lambda	N/A	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	Y	Y	Y	Y	Y
full block scope	Y	N	Y	Y	Y	Y	Y	Y	M	Y	Y	M	Y	Y	Y	Y	Y
downward funargs	Y	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
upward funargs	Y	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
algebraic datatypes	Y	N	N	Y	M	N	Y	Y	N	N	N	N	Y	Y	Y	N	Y
pattern matching	Y	N	N	N	M	M	Y	Y	N	N	N	N	Y	Y	Y	M	Y
expression *if*	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
expression *case*	Y	N	N	M	Y	Y	Y	Y	N	N	Y	N	Y	Y	Y	Y	Y
expression block/let	Y	N	M	M	Y	Y	Y	Y	M	M	Y	N	Y	Y	Y	Y	Y

Y	Yes, supported
M	Meh, somewhat supported
N	No, not supported
N/A	Does not apply

Figura 1: Características del paradigma funcional en los lenguajes de programación. [6]

2.1.4. Ventajas y Desventajas

Ventajas	Desventajas
El programa es mas expresivo al definir el programa con un enfoque en que es lo que quiere lograr a diferencia de como lo quiere lograr.	Es mas complicado generar buen código estando influenciado por el predominante pensamiento imperativo.
Tiene una consideración en la seguridad con la inmutabilidad asegurándonos que una variable con un valor asignado no se alterará.	Las referencias de variables que solo contienen resultados parciales generan trabajo extra para el colector de basura.
El orden de ejecución de las funciones puede ser cambiado o incluso se puede llegar a aplicar paralelismo cuando no exista dependencia de datos entre dichas funciones.	Ciertos elementos inmutables como estructuras de datos que son mutables en lenguajes imperativos no siempre tienen un eficiente homologo en el paradigma funcional.

Cuadro 1: Ventajas y desventajas del paradigma funcional.

2.1.5. Aplicaciones

Se ha usado este paradigma en una serie variada de aplicaciones a nivel industrial y académico. La mayoría de estas se enfocan en el tratamiento de colecciones de datos, como es el caso de analizar métricas de rendimiento y poder realizar filtrados personalizados de una manera clara y concisa, otra razón por la cual es preferido para el tratamiento de datos en forma especifica es por el potencial de escalabilidad, al poder identificar fácilmente las dependencias entre las tareas o las que son independientes, lo que permite explotar los recursos de la computadora como los múltiples procesadores y paralizar la carga de trabajo alcanzando mayores velocidades. [7]

En la educación se utiliza mayormente por lo intuitivo que es definir una función matemática y poder pasar esa forma de abstracción a la programación permitiendo una enseñanza mas fluida en temas de programación y desarrollo en computación.

2.2. Paradigma Lógico

2.2.1. Características

En el paradigma lógico el programa esta escrito en un lenguaje lógico, el cual se diferencia de un lenguaje por que ademas de tener una sintaxis y una semántica tiene reglas de inferencia, haciendo que la ejecución del programa sea un proceso de validación de un teorema. En otras palabras las operaciones que realiza son inferencias lógicas. [8].

Sintaxis Estas son las reglas que definen como escribir formulas lógicas.

Semántica Define el significado de las formulas en términos de consecuencias lógicas.

Reglas de inferencia Describen la forma correcta de llegar a una conclusión.

2.2.2. Metodologías

Algunos de los puntos que se deben enfatizar en el paradigma lógico para diferenciarlo de otros paradigmas son los siguientes: [9].

- Las operación se realizan en el dominio de todos los términos definidos en un alfabeto universal.
- Los valores son asignados a variables en sustituciones generadas automáticamente llamadas *unificadores generales*, dichos valores pueden contener variables llamadas *variables lógicas*.
- El control es manejado por un solo mecanismo: *backtracking automático*. Es decir que el control de flujo esta determinado por el seguimiento previo en las inferencias realizadas.

Otras capacidades que tiene este paradigma son:

Programación declarativa nos permite dar una *interpretación de procedimientos*, la cual describe el como las operaciones son realizadas, y una *interpretación declarativa* la cual describe el significado de que es lo que se esta computando. Lo cual permite entonces que al ejecutar un programa se esta ejecutando una especificación fácil de entender y desarrollar.

Programación interactiva lo cual es que para un solo programa el usuario puede interactuar con el de distintas formas realizando preguntas que le son de interés. A lo cual si es necesario el desarrollador puede poner puntos de seguimiento en el ciclo de interacción.

2.2.3. Lenguajes

Algunos de los lenguajes en el paradigma lógicos son [10]:

Absys

Alice

ASP (Answer Set Programming)

CycL

Datalog

Datomic

Logtalk

Mercury

Oz

Parlog

Planner

PROGOL

Prolog

2.2.4. Ventajas y Desventajas

Ventajas	Desventajas
Simple y conciso.	Uso de un solo mecanismo de control y un solo tipo de dato. (En algunos lenguajes puede modificarse)
Usos múltiples de un solo programa.	No tan eficiente como C.
El uso de backtracking puede ser de gran utilidad con problemas NP completos.	No ocupa un lugar reconocido por la industria a nivel de Java o C.

Cuadro 2: Ventajas y desventajas del paradigma lógico. [9] [8]

2.2.5. Aplicaciones

Mayormente las aplicaciones que tiene prolog son en el desarrollo de sistemas que interpretan otros lenguajes como el interprete del lenguaje Erlang, así como en el procesamiento de lenguaje natural, esto por la facilidad que tiene al parsear la entrada que en estos casos es un lenguaje completo. Una característica que se explota en sus aplicaciones es la de poder relacionar las reglas con consecuencias utilizadas en sistemas expertos con acceso a una gran base de conocimiento. Adicionalmente al tener soporte a multiples interacciones con un solo programa es utilizado en inteligencia artificial para la inferencia de tareas a realizar en un entorno cambiante. [11]

3. Conclusiones

El paradigma funcional y el paradigma lógico tienen casos de uso que hacen sus respectivos lenguajes que los representan una mejor selección que si seleccionaran algún otro con menos capacidades para las tareas a desarrollar. La información que respalde nuestras decisiones como desarrolladores debe estar fundamentada con una evaluación de lo que buscamos resolver y como esta puede beneficiarse de la mejor opción, si la definición del problema o la opción carece de beneficios al caso de uso el proyecto este será propenso de compensar sus carencias con complicaciones en el desarrollo.

Los paradigmas mostrados en este trabajo son solo algunos de los que existen actualmente y de los que existirán en cuanto la interrelación entre algoritmos y datos evolucione con el tiempo dándonos la tarea de seguir evaluando y creando mejores soluciones que se adecuen a los retos actuales.

Referencias

[1] L. Joyanes Aguilar, L. Rodríguez Baena, y M. Fernández Azuela, *Fundamentos de programación libro de problemas*. McGraw-Hill, 2008, OCLC: 629802677.

[2] R. Toal. Programming paradigms. [En línea]. Disponible en: https://cs.lmu.edu/~ray/notes/paradigms/. [Accedido: 14-8-2019].

[3] J. Hughes, "Why functional programming matters," *The computer journal*, vol. 32, no. 2, Noviembre 1989.

[4] V. Bera. (2015, Febrero) Functional programming - a new paradigm. [En línea]. Disponible en: https://www.hackerearth.com/practice/notes/functional-programming-a-new-paradigm/. [Accedido: 14-8-2019].

[5] Leandro. (2018, Noviembre) An introduction to the basic principles of functional programming. [En línea]. Disponible en: https://www.freecodecamp.org/news/an-introduction-to-the-basic-principles-of-functional-programming-a2c2a15c84/. [Accedido: 14-8-2019].

[6] J. Tov. (2018, Abril) What are some examples of functional programming languages? [En línea]. Disponible en: https://qr.ae/TWrAfS. [Accedido: 14-8-2019].

[7] J. Mott *et al.* Testimonials and quotes. [En línea]. Disponible en: https://fsharp.org/testimonials/. [Accedido: 14-8-2019].

[8] Y. Jia-Huai. Introduction to logic introduction to logic programming. [En línea]. Disponible en: http://www.eng.ucy.ac.cy/theocharides/Courses/ECE317/Logic%20Programming%201.pdf. [Accedido: 14-8-2019].

[9] K. R. Apt, "The logic programming paradigm and prolog," *CoRR*, vol. cs.PL/0107013, 2001. [En línea]. Disponible en: http://arxiv.org/abs/cs.PL/0107013

[10] C. H. (2019, Junio) Logic programming. [En línea]. Disponible en: https://www.computerhope.com/jargon/l/logic-programming.htm. [Accedido: 14-8-2019].

[11] Y. Wang. (2015, November) What are the applications of logic programming? [En línea]. Disponible en: https://qr.ae/TWrCJb. [Accedido: 14-8-2019].