

UN ENFOQUE DE LA SOLUCION DE PROBLEMAS EN LA ENSEÑANZA DE LA PROGRAMACION FUNCIONAL

An approach to the solution of the problems in the teaching of functional programming

RESUMEN

Este documento presenta una visión de cómo enseñar a los estudiantes a programar (programación funcional) siguiendo el ciclo de Polya. Se presentan cuatro pasos de un ciclo que enseña como programarlo y los pasos tomados son una muestra del que enseña Polya al plantear y resolver problemas. El lenguaje propuesto aquí es el llamado Haskell y el cual será presentado mediante ejemplos.

PALABRAS CLAVES: Ciclo de Polya, Lenguaje Haskell, programación funcional.

ABSTRACT

This document presents a vision of how to teach students programming (functional programming) following the Polya cycle. We present four steps of a cycle that teaches how to program and the steps taken are an example of the teacher Polya to raise and solve problems. The language proposed here is called Haskell and which will be presented by example.

KEYWORDS: Polya cycle, Haskell language, functional programming.

PEDRO PABLO CÁRDENAS ALZATE

Licenciado en Matemáticas y Computación.

M.Sc. Enseñanza de la Matemática.

Profesor Asistente

Universidad Tecnológica de Pereira

ppablo@utp.edu.co

FERNANDO MESA

Licenciado en Matemáticas y Física.

M.Sc. Instrumentación Física

Profesor titular

Universidad Tecnológica de Pereira

femesa@utp.edu.co

CARLOS ALBERTO RODRIGUEZ VARELA

Licenciado en Matemáticas y Física

Estudiante Maestría en Enseñanza

de la Matemática

Profesor Asistente

Universidad Tecnológica de Pereira

car@utp.edu.co

1. INTRODUCCIÓN

Muchos estudiantes presentan pocas dificultades al enfrentarse a la programación funcional, pues la mayoría de veces ellos tienen bases en programación y es esto lo que les hace que no presenten problemas a la hora de enfrentarse a la programación declarativa como es el ejemplo del lenguaje Haskell.

El problema ahora es como responder a los estudiantes la siguiente pregunta: ¿por dónde se empieza?. A través del libro de Polya: *Como plantear y resolver problemas* podemos responder en parte a este interrogante que los estudiantes tienen al enfrentarse cuando desean desarrollar un programa (diseñar un programa en Haskell). Cuando se intenta desarrollar un algoritmo, la pregunta es la misma que se hicieron los estudiantes de por dónde empezar, como resolverlo y es aquí donde

entra la relación que expone Polya en su famoso libro ya mencionado. Los pasos son muy similares, pues cuando se desea programar algo, en realidad se está resolviendo un problema.

2. COMO PROGRAMARLO

Al igual que el ciclo de Polya posee unas etapas las cuales nos conducen a resolver problemas, se presentan aquí cuatro etapas muy similares que le ayudan al estudiante a programar funcionalmente. Las etapas son:

- Entender el problema
- Diseñar el programa
- Escribir el programa
- Reflexión o visión retrospectiva

Entender el problema. Se debe entender el problema, nombrar o dar nombre al programa o función, cual es el tipo.

Diseñar el programa. En el diseño del programa, se necesita pensar acerca de las conexiones entre las entradas y salidas. Si no existe una conexión inmediata, se debe pensar en problemas similares ya resueltos los cuales ayudaran en la solución.

Escribiendo el programa. Utilizar el lenguaje de programación adecuado y que sobre todo incluyendo el mismo lenguaje sea suficiente. Utilizar la mayor recursividad posible, pues el éxito de un buen programa es su suficiencia y que sea siempre pensando en el público.

Reflexión. Examinar la solución, observar si se utilizaron variedad de argumentos al programarlo. Qué tipo de funciones o procedimientos se pueden mejorar, cuales sobraron y como reemplazarlos para su eficiencia.

En el diseño de cualquier programa, el estudiante que tiene presente tan solo estas cuatro etapas, no presentará problemas mayores para lograr los objetivos esperados. El lector puede verificar que el ciclo de cuatro pasos es similar al ciclo de Polya.

3. PROGRAMANDO EN HANSKELL

Obsérvese un ejemplo sencillo de cómo se desarrollará un programa que encuentra el máximo de tres enteros siguiendo el ciclo nombrado anteriormente.

- Problema: Encontrar el máximo de tres números enteros.
- Entendiendo el problema: ¿Qué hacer cuando dos o tres enteros son el máximo?
- Diseñando y escribiendo el problema: Aquí se presentan puntos interesantes como los siguientes: diseñar el programa que compare tres enteros el usuario podría hacer un código de líneas cortas y otros procedimientos más eficientes.
- Reflexión: ¿Existirá otra forma de realizar este programa, que pasaría si en vez de tres números fueran más?. Finalmente el usuario pudo haber hecho una función que puede para muchos ser más eficiente que otras y en pocas líneas. Podría seguirse presentando más ejemplos, los cuales mediante el ciclo de los cuatro pasos se programarían sin dificultad como son: hallar el máximo de una lista, decidiendo cuando una lista es una sublista, etc.

4. CASO DE ESTUDIO: LOS PALINDROMOS

El problema de reconocer palíndromos se presenta con el siguiente ejemplo: **anitalavalatina**. Este es un problema que ayudan a los estudiantes a tener que desarrollar buenas estrategias de pensamiento para poder programarlo y que sobre todo sea lo mayor eficiente posible. Se va a seguir de nuevo los cuatro pasos del ciclo como programarlo.

- Entendiendo el problema. El estudiante debe saber o identificar que palíndromo es una cadena la cual se puede leer de derecha a izquierda y viceversa, pero teniendo presente que los espacios en la cadena y signos de puntuación. Las letras mayúsculas y tipos de letra. Obsérvese que lo importante de este paso del ciclo, pues se debe tener en claro los que el problema dice y a lo que el estudiante (en particular) debe enfrentarse.
- Diseñando el programa. Existen en verdad dos formas de diseñar, una de ellas es diseñar el caso más simple, es decir, palíndromos no tan largos y posterior a la evaluación de este, diseñar los casos generales.
- Escribiendo el programa. Existen en verdad muchas formas de como los estudiantes programarían este problema de los palíndromos. Aquí lo que se trata es de hacerles ver a ellos que entre menos procedimientos tenga que realizar el programa, mejor eficiente será. El puede utilizar listas, arreglos e inclusive puede responder booleanamente o con otro tipos de respuestas.
- Reflexión. Como mejorar el diseño del programa para que acepte todo tipo de letras, inclusive signos de puntuación. Aprovechar la recursividad (programa declarativo) para mejorar procedimientos que hagan más eficiente el programa, pues esto es lo que hacen un buen programa.

5. VISIÓN DE LA EDUCACIÓN EN TECNOLOGÍA

Si en un futuro se pretende una enseñanza en la solución de problemas a través de medios tecnológicos que contribuya a alcanzar *competencias* para la vida académica, necesitamos entonces desarrollar en los estudiantes capacidades aplicables a situaciones nuevas y sobre todo la Globalización. La Educación en Tecnología a través de la programación funcional puede aportar significativamente para:

Aprender a Aprender y Aprender a Pensar

Muchas técnicas cambian hoy vertiginosamente, pero un factor puede y debe mantenerse: la habilidad de los estudiantes para resolver problemas y para pensar críticamente a la tecnología.

Se pretende a que los estudiantes conozcan mejor el mundo que los rodea y adquieran cierto tipo de aptitudes de control sobre la Resolución de problemas. Por lo tanto, no interesan tanto los problemas en sí sino el desarrollo de las capacidades complejas que se adquieren a partir de la resolución de los mismos.

Siempre se ha reconocido a la *experiencia* como una fuente de conocimiento, pero la enseñanza de la resolución de problemas utilizando la programación funcional es muy valiosa, puesto que es mucho más fácil comprender los procedimientos, su racionalidad y sus resultados. Por este motivo, el trabajo en la resolución de problemas implica un espacio individual y un trabajo grupal en equipo.

6. REFLEXIÓN

El diseño de algoritmos para la solución de problemas utilizando la programación funcional es la base de las ciencias de la computación. Un Enfoque de programación funcional enseña las habilidades necesarias para dominar este tema esencial. Hoy en día muchos especialistas rechazan los métodos más tradicionales de los algoritmos de la enseñanza mediante el uso de un contexto de programación funcional con Haskell como lenguaje base.

Lo anterior conduce a que el programador entienda el algoritmo en sí más rápidamente y poder usar ese conocimiento para explorar soluciones alternativas al problema en cuestión.

El objetivo es enseñar a tratar secuencias adecuadamente, es decir en el estudio del concepto de secuencia y del buen tratamiento en el uso de las estructuras iterativas para su correcto funcionamiento. Es por ello que se recalca la importancia de la enseñanza a través de algoritmos tal y como lo hacer Polya.

Así, el objetivo fundamental es minimizar los errores que se comenten cuando usamos iteraciones. Por ello presentamos los esquemas algorítmicos fundamentales propuestos en [2]. Ellos parten de la idea que las distintas

soluciones de los problemas en cuestión tengan un patrón en común, que comúnmente se denominan esquema, y los cuales funcionan como una plantilla o guía para encontrar un algoritmo que resuelva el problema.

Los esquemas son un método utilizados para construir algoritmos que se fundamentan en el hecho de que toda iteración, leída a posteriori, puede ser interpretada como la enumeración de una serie de acciones. En tal listado se accede a cada uno de los objetos de la secuencia y, para cada uno de ellos, se aplica un tratamiento único, dependiendo básicamente del objeto y del problema a resolver.

Es entonces con base a lo anterior que un problema matemático puede resolverse por etapas; cada una de ellas tiene una solución que puede llegar a ser diferente a la siguiente, y es entonces cuando se anidan todas ellas para formar la solución final.

Esto se basa en general en dos hipótesis: la entrada a los elementos puede ser tomado en término de las acciones: inicialización de la adquisición de los elementos y obtener siguiente elemento, y el tratamiento de los elementos viene descrito en términos de las siguientes acciones: inicialización del tratamiento, tratamiento del elemento corriente y tratamiento final. Se reconocen tres tipos de problemas sobre secuencias que abarcan a toda la familia de problemas:

Recorrido que denominaremos (R), la búsqueda que será (B) y recorrido parcial (RP), que es el equivalente a: recorrido + búsqueda. A su vez, dentro de los problemas de recorrido se distinguen el esquema para el tratamiento integrado de la secuencia vacía y del primer elemento (R1), el esquema para el tratamiento especial de la secuencia vacía (R2) y el esquema para el tratamiento especial del primer elemento (R3).

Esta familia de esquemas relacionada con lo mencionado anteriormente referente a que un problema se puede resolver por etapas, se aplica a dos tipos de secuencias: secuencias con marca inicial y secuencias con marca final. Cada tipo de secuencias tiene sus propios esquemas.

Es aquí donde muchos expertos difieren a la hora de aplicar la programación funcional para la resolución de problemas; ello conlleva a que se pierda interés en la enseñanza de la programación y en cambio se regrese a lo tradicional a la hora de solucionar un problema. La heurística es uno de los elementos fundamentales para que el estudiante adquiera esa destreza que debe poseer para aplicar la programación funcional en la obtención de resultados de un problema en cuestión.

El programa de los cuatro ciclos es esencial a la hora de abordar un problema de cualquier índole de tipo

académico, ya que debido a su estructura funcional, este se adapta de manera correcta a lo que el estudiante persigue, brindándole entonces pautas necesarias a la hora de plantear posibles soluciones.

Cabe resaltar que es poco el profesorado que aplica dicha técnica, ya que la programación funcional no ha sido del todo enseñada y por ende se recurre a métodos tradicionales y un poco obsoletos que hacen que el estudiante no preste el interés necesario y tenga temor enfrentarse a un desafío como es el de resolver un interrogante: el problema.

Podemos definir que la resolución de problemas a través de la programación funcional como un proceso en el cual el estudiante concibe lo irracional. En tal diseño podemos distinguir dos fases; una de ellas es el esquema mental: representación mediadora entre el sujeto y el entorno; la otra fase es ya algo concreto: el plan proyectivo que ordena los recursos y actúa para lograr un producto determinado.

Por consiguiente, el diseño es la actividad fundamental en el proceso de creación de cualquier algoritmo de programación, lo que involucra tanto la imagen mental inicial del mismo como la representación gráfica, los diagramas y dibujos, el modelado y la programación de los pasos a seguir.

Estos procesos de diseño son la base de las operaciones técnicas que los estudiantes realizan, replicando así una característica de los: la habilidad de la humanidad a través de la historia de explorar, entender y sobrevivir en un ambiente determinado y enfrentarse así a resolver los problemas del entorno.

5. CONCLUSIONES

Este artículo enseña la forma de enseñar programación funcional mediante un ciclo semejante al de Polya, es más, se basó en Polya para implementarlo en los laboratorios de programación funcional. Este ciclo enseña a los estudiantes como enfrentarse a un programa por más complejo que este parezca y algo importante, es que se puede aplicar a cualquier lenguaje de programación.

Lo importante acá es que se mostró que el programa de los cuatro ciclos sirve como guía para que el estudiante no solo lo aplique en programación, si no en cualquier problema de su carrera profesional.

BIBLIOGRAFÍA

[1] G. Polya. Como plantear y resolver problemas. Editorial Trillas, 2002.

[2] R. Bird. Introducción a la programación funcional con Haskell. Editorial Prentice. Segunda Edición, 2000.

[3] H. Balbuena. La enseñanza de las Matemáticas en la escuela. S.E.P. México, 1995.

[4] I. Fuenlabrada. Juega y aprende Matemáticas. S.E.P. México, 1994.

[5] teyet-revista.info.unlp.edu.ar