

UNIVERSIDAD NACIONAL COSTA RICA

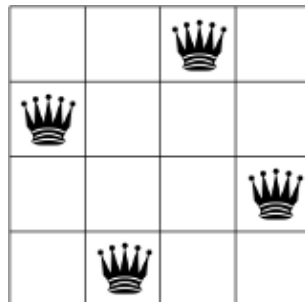


Facultad de Ciencias Exactas y Naturales

Curso:

Paradigmas de Programación

Proyecto 1: NQueens



Profesor:

Eddy Miguel Ramírez

Estudiantes:

Mario Arguello
Josué Céspedes Miranda

II CICLO

2020

Índice

1. Resumen ejecutivo	2
2. Introducción	3
3. Marco teórico	4
4. Descripción de la solución	5
5. Resultados de las pruebas	6
6. Conclusiones	7

1. Resumen ejecutivo

El presente trabajo esta realizado con el objetivo de hacer un uso práctico de distintos modelos con el fin de brindar una solución al popular problema NReinas bajo el uso de un paradigma funcional, que permita conocer nuevos conceptos y permita establecer diferentes perspectivas a la hora de brindar soluciones a distintos problemas mediante programación.

El problema de NReinas es un acertijo con bastante popularidad, propuesto por el ajedrecista alemán Maxx Bezzel en 1848, consiste en colocar N reinas en un tablero con dimensiones $N \times N$, su principal desafío radica en colocar las N reinas, mencionadas anteriormente, de tal manera que ninguna reina quede atacando otra. Dado que en el ajedrez esta pieza puede atacar a las piezas que se encuentren en su misma fila, misma columna, o en sus respectivas diagonales.

En este documento se detallan dos posibles métodos para dar solución al problema explicado previamente. La primera, y más común a la hora de resolver este algoritmo, es una solución implementando backtracking, esta solución se ha diseñado y escrito en Scheme, el cuál como se mencionó al inicio tiene características de programación funcional y es un subconjunto del lenguaje de programación LISP; y que fue desarrollado en el MIT (Massachusetts Institute of Technology). La segunda solución ha sido implementada sobre algoritmos genéticos, se ha trabajado con elitismo y con un porcentaje de mutación de un 5 %, la solución para este caso esta escrita en Erlang, el cuál es también es un lenguaje de programación funcional de alto nivel, diseñado en los años 80 en los laboratorios de Ciencias y Computación de la compañía de telefonía sueca Ericsson, y fue diseñado para escribir aplicaciones concurrentes y distribuidas de funcionamiento ininterrumpido.

Además, este trabajo pretende plasmar las distintas diferencias presentes en los métodos utilizados para dar solución al problema, establecer una comparación basada en el tiempo de ejecución de cada una de las soluciones y que permita obtener como resultado cuál presenta mejores características de desempeño. Por otra parte, ambas soluciones representan una oportunidad para los estudiantes participantes en conocer nuevos paradigmas, poder incursionar en temas de mucho interés como en el caso de algoritmos genéticos que son temas muy amplios y que lamentablemente es difícil encontrar en un curso dentro del programa de la carrera de Ingeniería en Sistemas de Información en la Universidad Nacional de Costa Rica que abarque la investigación e implementación de este tipo de algoritmos, que se encuentran relacionados a temas de inteligencia artificial y que son utilizados para resolver problemas de búsqueda y optimización, que concuerdan con los parámetros necesarios para dar una solución al problema de NReinas. Por lo que este proyecto representa: 1) un reto, por la parte de investigación y la utilización de un paradigma distinto a la programación orientada a objetos(POO), que incluye un cambio en la forma de pensar, y 2) la oportunidad de aprender temas nuevos y relevantes que pueden llegar a tener un impacto positivo en un futuro profesional.

Aquí se documentan la descripción de las soluciones, los principales problemas encontrados durante su desarrollo, aprendizajes obtenidos, se adjuntan los resultados de las pruebas efectuadas y demás aspectos concernientes para el proyecto.

2. Introducción

3. Marco teórico

Como se ha mencionado previamente, para dar solución al problema de NReinas se han utilizado dos lenguajes de programación funcionales. A continuación se profundiza un poco más en la historia, características de Scheme y Erlang y sus principales usos.

1. Scheme

Es un lenguaje de programación de alto nivel, se le considera un lenguaje altamente expresivo. Su sintaxis es fácil de leer pero no lo es tanto comparándolo con Python. Opera sobre estructuras como listas, cadenas, vectores, tuplas y posee conjuntos de datos que lo hacen muy versátil.

- Historia: surgió en los laboratorios del MIT en 1975, cuando Guy L. Steele y Gerald J. Sussman buscaban un lenguaje con una semántica clara y sencilla. Originalmente se llamaba "Schemer", se le cambió el nombre ya que sus creadores usaban el sistema operativo ITS, que limitaba la longitud del nombre de los ficheros a 6 caracteres.
- Antecedentes: altamente influenciado por el cálculo lambda. Su desarrollo ha sido lento, ya que las personas encargadas de su estandarización son muy conservadoras y ha sido difícil añadirle nuevas características. Se le considera uno de los mejores lenguajes de propósito general. Fue el primer dialecto de Lisp que usó ámbito estático o léxico en lugar de dinámico de forma exclusiva.
- Características: filosofía minimalista, su mecanismo principal para el control de flujo son las llamadas recursivas finales. Además, Scheme ofrece gestión automática de memoria, popularmente conocido como recolector de basura. Las listas son la estructura de datos básica, no hay sintaxis explícita para crear registros o estructuras o para programación orientada a objetos. La programación funcional pura no necesita de variables ni manejo de memoria compartida, por lo que hacen a Scheme una buena opción en presencia de procesos concurrentes. Una de las desventajas es que existen múltiples implementaciones diferentes, cada una utiliza extensiones y bibliotecas propias que se tornan incompatible entre sí.
- Usos: la mayoría de uso que se le da a Scheme se refiere a la parte educativa. Profesores de universidad, como es el caso del profesor R. Kent Dybvig que trabaja en la Universidad de Indiana, se concentra en el prototipado de nuevas semánticas para Scheme, incluso tiene a la venta un compilador de Scheme con su colega Daniel P. Friedman. También, hay muchos otros profesores de distintas universidades que lo usan como lenguaje de enseñanza.

2. Erlang

Es un lenguaje de programación funcional de alto nivel, que se ubica también dentro del paradigma de Programación Declarativa, diseñado para escribir aplicaciones concurrentes y distribuidas de funcionamiento ininterrumpido. Erlang cuenta con tipado dinámico y asignación única, esto último consiste en que una vez asignado un valor a una variable, éste valor no puede cambiar.

- Historia
- Antecedentes
- Características

- Usos
- Bibliotecas utilizadas

4. Descripción de la solución

5. Resultados de las pruebas

6. Conclusiones