Instituto TecnolÃ^3*gicodeCostaRica*

Ãrea de IngenierÃa en Computadores

Lenguajes, Compiladores e IntÃ©rpretes

Profesor: Kevin Moraga

Estudiantes: Victor Montero
NicolÃ¡s JimÃ©nez

Proyecto II: Acertijo en Prolog

I Semestre, 2015

Problema

El problema propuesto, fue el de construir un programa computacional eficiente capaz de resolver un rompecabezas constituido por una figura y varias piezas provistas por el usuario, estas piezas deben ser acomodadas dentro de la figura de modo de que calcen por completo. El programa ademÃ¡s debe probar el acomodamiento de piezas girÃ¡ndolas en 90 grados, 180 grados y 270 grados. Finalmente dada la caracterÃstica de Prolog de ser no determinÃstico (significa que no garantiza una Ãºnica soluciÃ^3*n ytampoco garantizaqueexistaalgunasoluci*^{3}*n*), *elprogramadebemostrarcadaposiblesoluci*^{3}*nsiexiste, envezdeu*

La forma en que se representan la figura y las piezas es con una lista de listas en donde los elementos se representan con una x, y en donde los elementos vacÃos se representan con un o. Por Ãºltimo en la pregunta que se hace al programa se incluye ademÃ¡s una variable sin instanciar. En esta variable no instanciada el programa intenta encontrar el o los valores correspondientes para que

la pregunta resulte con un valor booleano verdadero. Este proceso de intentar encontrar el valor que implique que la pregunte se convierta en verdadera se llama unificaciÃ³n. La unificación es una característica notable de los lenguajes de programación del paradigma lÃ³gico.

El problema propuesto conlleva varias implicaciones. Primero se debe aprender y comprender con denuedo el funcionamiento del paradigma lÃ³gico, el cual simboliza un gran reto. Este proyecto adem s envuelve varios de los con la unificación, reglas y hechos, recursividad, manejo de estructuras lógicas y los cuts.

Para poder resolver este problema se necesitan cubrir una serie de pasos que llevan directamente al resultado final. A continuaciÃ³n se explicar con densada mente la estrategia de solución usada en particular por nosotros para apo

El primer paso que se efectuÃ³ fue generar todas las posibles soluciones que se pueden tener para rellenar la figura con las pi Con posibles soluciones nos referimos a encontrar y almacenar todos los estados posibles de la figura, y esto significa : todas las posibles combinaciones de posiciones en la figura con todas las combinaciones de rotaciones posibles (0 grados, 90 grado

El segundo paso fue el de crear una funciÃ³n o regla capaz de manipular las matrices de fichas para lograr los giros de 90 gra posteriormente poder comparar las fichas con la figura de manera correcta.

El tercer paso necesario que se realizÃ³ fue el de comparar si las fichas al menos caben dentro de la matriz de la figura, de esta o, lo cual implica que sea un espacio idóneo para la colocación de la ficha.

El Ãºltimo paso y ademÃ¡s el mÃ¡s complicado es el de intentar colocar las piezas en la figura de manera que este sea completado o llenado, para realizar esto se toman las soluciones que pasaron por el filtro anterior y se intentan colocar en la figura contemplando cada rotaciÃ³n y posible ubicación en las coordenadas de la figura

Ambiente de Desarrollo

En esta secciÃ³n se comenta acerca del ambiente de desarrollo utilizado para la construcción del programa de prolog, asimi

Para el desarrollo del programa se escogiÃ³ el ambiente de programación SWI-Prolog. El cual es una implementación de c adengran funcionalidad allenguaje. Ademsdeposeercapacidaddeinterfazgrficalocuallohacenico. Finalmenteincluyeunac como ca dam©todo.

El SWI-Prolog es el compilador utilizado para programar, sin embargo el cÃ³digo se escribe en el editor de texto : Sublime Text 2, el cual incluye herramientas que hacen el proceso de escritura de código sea migable, como resaltado de texto til p

Prolog es un lenguaje de programaciÃ³n del paradigma lógico, de hecho es el ms conocido y utilizado, siendo por lo tanto el len

La base de conocimiento: Esta parte del lenguaje es donde se escribe el programa para luego ser compilado y ejecutado. Para poder crear la base de conocimiento se puede utilizar cualquier editor, como se mencionÃ³ anteriormente se utiliz el editor Sublime Text 2.

La parte de preguntas, Prolog es un lenguaje de queries(preguntas), en donde se intentan conseguir valores de verdad. En esta parte fue donde se utilizÃ³ SWI−Prolog, en esta parte ademses donde es tel compilador y donde sem
Por Ãºltimo en Prolog aunque es complejo, consiste en bÃ¡sicamente dos estructuras para su funcionamiento: los hecho y las reglas. Los hechos son la estructura mÃ¡s simple de Prolog y sirven para indicar una relaciÃ³n siempre verdadera entre dos o mstomos. Por otro lado las reglas son la otra estructura importante, esta es ir

Estructuras de datos usadas

En esta secciÃ³n se explican las estructuras de datos utilizadas en general y además como se representó y estructuró la figura como las distintas fichas que lo componen. Por ltimo se habla de la forma en que se manipulan las dichas estructuras.

Como estructura de datos bÃ¡sica se utilizÃ³ las listas que Prolog tiene por defecto, las cuales fueron usadas vastamente y

Para almacenar la figura del rompecabezas y las demÃ¡s figuras que el programa utiliza se usÃ³ listas de listas, ya que su eficacia radica en el almacenamiento y manejo de estructuras bidimensionales como matrices. Pr sucesivamente.

Como parte inicial de la soluciÃ³n de este problema, se crearon gran cantidad de soluciones de piezas las cuales incluyen cad

Se explicarÃ¡ ahora un poco acerca de esas coordenadas anteriormente mencionadas. Estas coordenadas representan la posiciÃ³n de una pieza en la figura del rompecabezas, el cual es una matriz bidimensional. Las posiciones c $1 y y = 1$, y se desplazan hacia la derecha y hacia abajo en una nym cantidad de filas y columnas.

Instrucciones para ejecutar el programa[2028?]

Para poder ejecutar este programa se deben satisfacer unos pocos requerimientos, los cuales se listan a continuaciÃ³n :

-Tener una computadora con un sistema operativo GNU-Linux, como Ubuntu.

-Tener instalado SWI-Prolog, para poder montar el programa y compilar.

DespuÃ©s de cerciorarse que se satisfacen las condiciones, se debe ejecutar el programa SWI-Prolog, despuÃ©s montar el archivo del programa de Prolog, para realizar esto se debe proveer su ubicaciÃ³n absoluta dentro de la computadora.

En este punto ya todo esta listo para proseguir con la pregunta la cual es como se ejecuta el procedimiento para finalmente obtener las soluciones de parte del programa. Para hacer la pregunta simplemente se debe escribir una instrucciÃ³n de este tipo : $figura(RepresentacionFigura, ListaPiezas, Solucion)$.

En donde representaciÃ³n figura es una matriz con todas los elementos que trae consigo la figura del rompecabezas. Y Lista

Corridas de Ejemplo

Entrada: figura([[x,x,x],[x,o,x]], [a, [x,x] ,[x,x]], [b, [x],[x]]], Sol).

Resultado: Sol=[(a,90,1,1), (b,0,1,3)].
Sol=[(b,0,1,1), (a,180,1,2)].

DescripciÃ³n de las principales relaciones

FALTA

Para la relaciÃ³n con la que se mueve una pieza se utiliza una reglaque toma una matriz y la rota a 90 grados, 180 grados o 270 g

Un mÃ©todo Ãºtil para rotar una matriz es primero transponerla y luego intercambiar las columnas. Para obtener la matriz transpuesta se utilizÃ³ $elm©todotransposedelabibliotecadeSWI - Prologclpfd$.

Al obtener la matriz transpuesta, finalmente solo se debe revertir el orden de todos los elementos de cada sub lista de la matriz, para poder realizar esto se utilizo el mÃ©todo reverse de prolog.

Al hacer esto ya se obtiene una matriz completamente rotada en 90 grados. Si se desea la rotaciÃ³$nde180gradossolosedeb$

Comentarios Finales

Conclusiones

- Se comprobÃ³$queelparadigmadeprogramaci^3nl^3gicoesrealmentetilensituacionesderesolverproblemasrecursivosyp$

- Se concluye a partir del proyecto que SWI-Prolog es posiblemente la mejor herramienta para construir software en Prolog por la gran cantidad de documentaciÃ³$n, bibliotecasygranconveniencia$.

- El paradigma de programaciÃ³$nesbastantetiladems para ampliarlascapacidadesdeprogramaci^3ndelingeniero, yaquet$

Problemas Encontrados

Un problema importante que se tuvo fue el desconocimiento agudo del paradigma y de su funcionamiento. Por lo tanto se debiÃ³$realizarunagraninvestigaci^3nparapoderasimilarlamayorpartedellenguajedeprogran$

Se tuvo gran dificultad comparar las fichas con las figuras, ya que el manejo de estructuras de matrices es bastante diferente a lo que se tenÃa anteriormente.

Otro problema que se enfrentÃ³$fuealahoradeusarelm©todoappendcondosvariablesnoinstanciadasyaqueproduc-agrancantidaddeerroresenlasoluci^3ndelaslistas$.
A modo de conclusiÃ³$nparaestasecci^3nelmayorretodificultadconelqueseconvivi^3fueeldecambiarelmododepensarau$