Práctica 6

Inteligencia Artificial
Grupo 05

Borja Aday Guadalupe Luis

Eduardo Bryan de Renovales Alvarado

Índice

1. Parentescos familiares.

1.1. Implementación del código de parentescos familiares.

En primer lugar, dada la representación de progenitores(progenitor1, progenitor2, descendiente, entendemos que los nombres de los padres pueden estar dispuestos en cualquier orden, por lo que para evitar repetición de código implementamos una función progenitor(X, Y) de tal forma que esta nos devuelva si X es padre o madre de Y.

Las funciones: padre(X, Y) y madre(X, Y) usarán el predicado progenitor(X, Y) y exigirán que X sea hombre o mujer dependiendo de si se busca el padre o la madre.

Las funciones: hijo(X, Y) e hija(X, Y) necesitarán comprobar que Y es progenitor de X (progenitor(Y, X) y que X sea hombre o mujer dependiendo de si buscamos hijos o hijas.

Las funciones: hermano(X, Y) y hermana(X, Y) primero comprobarán que haya un progenitor común entre X e Y tal que X sea distinto de Y, además de que X sea hombre o mujer dependiendo de lo que busca el predicado inicial.

Las funciones: abuelo(X, Y) y abuela(X, Y) comprueban que X sea padre o madre (dependiendo de si el predicado es abuelo o abuela) de un Z y que Z sea padre o madre de Y.

Las funciones: primo(X, Y) y prima(X, Y) comprueban que existan dos hermanos o hermanas Z1 y Z2 tal que Z1 o Z2 sea padre o madre de X y Z1 o Z2 sea madre de Y, exigiendo además que X sea hombre o mujer dependiendo del predicado primo o prima.

La función ascendiente(X, Y) se ha implementado de forma recursiva pues no se puede escribir en una sola regla ya que descendiente incluye muchos familiares atrás. Para implementarla hemos realizado un caso base que indica que X es padre o madre de Y. Si esto no se cumple, entra en el caso recursivo, que básicamente comprueba que haya un padre o madre Z de Y tal que X sea ahora ascendiente de Z. De esta forma va recorriendo el árbol genealógico hacia arriba hasta encontrar un parentesco entre X e Y.

1.2. Ejemplos de consultas en parentescos familiares con sus resultados.

Consulta 1: Saber quien es el padre de Pablo.

```
?- padre(X, pablo).
X = pedro ;
false.
```

Consulta 2: Saber quienes son los progenitores de Pablo.

```
?- progenitor(X, pablo).
X = pedro ;
X = rosa.
```

Consulta 3: Saber quienes son los hijos de Juan.

```
?- hijo(X, juan).
X = luis.
```

Consulta 4: Saber que hombres tienen hermanos o hermanas y quienes son.

?- hermano(X, Y).
X = luis,
Y = rosa ;
X = pablo,
Y = begona ;
X = luis,
Y = rosa ;
X = pablo,
Y = begona ;
false.

Consulta 5: Saber que personas son primas de alguien y quienes son.

?- prima(X, Y).
X = begona,
Y = miguel ;
X = begona,
Y = miguel ;
false.

Consulta 6: Saber quienes son los ascendientes de Miguel.

?- ascendiente(X, miguel).
X = luis ;
X = pilar ;
X = juan ;
X = maria ;
X = jose ;
X = laura ;
false.

2. Problema de las garrafas.

2.1. Implementación del código de las garrafas.

Lo primero que necesitamos para resolver el problema de las garrafas es proponer una representación del estado, indicar el estado inicial y los estados objetivos.

Decidimos que la representación del estado sea dos datos numéricos que indiquen la cantidad de agua que hay en cada momento en cada una de las botellas, en nuestro programa B1 será la botella de 4L y B2 la de 3L.

El estado inicial tendrá las dos botellas vacías estado(0, 0) y el estado objetivo será aquél en el que alguna de las botellas contenga exactamente 2L, estado(2, _) o estado(_, 2).

Una vez tenemos representados los estados iniciales y objetivos, procedemos a implementar las distintas operaciones que nos permiten pasar de un estado a otro:

- Llenar la botella X: comprueba que la botella X no está llena y si no lo está la llena poniendo su contenido a su máxima capacidad.

- Vaciar la botella X: comprueba que la botella no esté vacía y si no lo está pone su contenido a 0.
- Verter la botella X en la botella Y: Comprueba que la botella X no está vacía y la botella Y no está llena, calcula cuantos litros de agua quedará en cada una de ellas y actualiza su contenido.

Todas estas operaciones están por duplicado pues no es lo mismo vaciar, llenar o verter B1 o B2.

Finalmente se implementa una función consulta que no recibe parámetros la cual inicia el proceso de encontrar la solución y va devolviendo las soluciones que encuentra sin repetición de estados.

2.2. Ejemplo de consulta en problema garrafas.

?- consulta.

Visitados:
[estado(4,2),estado(3,3),estado(3,0),estado(0,3),estado(4,3),estado(4,0),estado(0,0)].
SOLUCION ENCONTRADA sin repetición de estados:
[Llenar B1,Llenar B2,Vaciar B1,Verter B2-B1,Llenar B2,Verter B2-B1]
Estados camino:
[estado(0,0),estado(4,0),estado(4,3),estado(0,3),estado(3,0),estado(3,3),estado(4,2)]
true ;

Visitados:
[estado(0,2),estado(4,2),estado(3,3),estado(3,0),estado(0,3),estado(4,3),estado(4,0),estado(0,0)]
SOLUCION ENCONTRADA sin repetición de estados:
[Llenar B1,Llenar B2,Vaciar B1,Verter B2-B1,Llenar B2,Verter B2-B1,Vaciar B1]
Estados camino:
[estado(0,0),estado(4,0),estado(4,3),estado(0,3),estado(3,0),estado(3,3),estado(4,2),estado(0,2)]
true ;

Visitados:
[estado(2,0),estado(0,2),estado(4,2),estado(3,3),estado(3,0),estado(0,3),estado(4,3),estado(4,0),estado(0,0)]
SOLUCION ENCONTRADA sin repetición de estados:
[Llenar B1,Llenar B2,Vaciar B1,Verter B2-B1,Llenar B2,Verter B2-B1,Vaciar B1,Verter B2-B1]
Estados camino:
[estado(0,0),estado(4,0),estado(4,3),estado(0,3),estado(3,0),estado(3,3),estado(4,2),estado(0,2),estado(2,0)]
true ;

Visitados:
[estado(2,3),estado(2,0),estado(0,2),estado(4,2),estado(3,3),estado(3,0),estado(0,3),estado(4,3),estado(4,0),estado(0,0)]
SOLUCION ENCONTRADA sin repetición de estados:
[Llenar B1,Llenar B2,Vaciar B1,Verter B2-B1,Llenar B2,Verter B2-B1,Vaciar B1,Verter B2-B1,Llenar B2]
Estados camino:
[estado(0,0),estado(4,0),estado(4,3),estado(0,3),estado(3,0),estado(3,3),estado(4,2),estado(0,2),estado(2,0),estado(2,3)]
true

3. Problema de las garrafas generalizadas.

3.1. Implementación del código de las garrafas generalizadas.

Para la realización de este problema nos hemos basado en el problema anterior y hemos añadido unos cambios para hacerlo generalizado.

- La nueva definición de estado objetivo es:
  objetivo(estado(B1, _), L):- B1 = L.
  objetivo(estado(_, B2), L):- B2 = L.

  Para comprobar si un estado es objetivo la botella1 o la botella2 debe contener exactamente los L litros que me pide la consulta.

- Las nuevas operaciones reciben ahora las capacidades de las botellas que sufrirán cambios, utilizan estos datos para realizar los cálculos de como quedará cada botella tras aplicar la operación. Seguimos teniendo las mismas operaciones que en el problema anterior con las mismas comprobaciones solo que ahora usamos como capacidades las que nos dan en la operación.

- Se ha cambiado el predicado consulta por consulta(B1, B2, L) pues ahora necesitamos que nos digan las capacidades de las botellas y los litros que se buscan y el predicado puede ya que necesita esos mismos datos para poder operar correctamente.

Por lo demás, el problema es exactamente igual al anterior. Podemos comprobar como con una serie de cambios muy pequeños, obtenemos un programa mucho más general con mayor utilidad que el anterior.

3.2. Ejemplo de consulta en problema garrafas generalizadas.

Consulta 1:

?- consulta(5, 7, 3).

B1 = 5, B2 = 7 y L = 3

Visitados:
[estado(5,3),estado(1,7),estado(1,0),estado(0,1),estado(5,1),estado(0,6),estado(5,6),estado(4,7),estado(4,0),estado(0,4),estado(5,4),estado(2,7),estado(2,0),estado(0,2),estado(5,2),estado(0,7),estado(5,7),estado(5,0),estado(0,0)]

SOLUCION ENCONTRADA sin repetición de estados:
[Llenar B1,Llenar B2,Vaciar B1,Verter B2- B1,Vaciar B1,Verter B2- B1,Llenar B2,Verter B2- B1,Vaciar B1,Verter B2- B1,Llenar B2,Verter B2- B1,Vaciar B1,Verter B2- B1,Vaciar B1,Verter B2- B1,Llenar B2,Verter B2- B1]

Estados camino:
[estado(0,0),estado(5,0),estado(5,7),estado(0,7),estado(5,2),estado(0,2),estado(2,0),estado(2,7),estado(5,4),estado(0,4),estado(4,0),estado(4,7),estado(5,6),estado(0,6),estado(5,1),estado(0,1),estado(1,0),estado(1,7),estado(5,3)]
true

Consulta 2:

?- consulta(12, 15, 3).

B1 = 12, B2 = 15 y L = 3

Visitados:
[estado(12,3),estado(0,15),estado(12,15),estado(12,0),estado(0,0)]

SOLUCION ENCONTRADA sin repetición de estados:
[Llenar B1,Llenar B2,Vaciar B1,Verter B2- B1]

Estados camino:
[estado(0,0),estado(12,0),estado(12,15),estado(0,15),estado(12,3)]
true

Consulta 3:

?- consulta(4, 8, 2).

B1 = 4, B2 = 8 y L = 2

false.

Consulta 4:

?- consulta(5, 8, 4).

B1 = 5, B2 = 8 y L = 4

Visitados:
[estado(5,4),estado(1,8),estado(1,0),estado(0,1),estado(5,1),estado(0,6),estado(5,6),estado(3,8),estado(3,0),estado(0,3),estado(5,3),estado(0,8),estado(5,8),estado(5,0),estado(0,0)]

SOLUCION ENCONTRADA sin repetición de estados:
[Llenar B1,Llenar B2,Vaciar B1,Verter B2- B1,Vaciar B1,Verter B2- B1,Llenar B2,Verter B2- B1,Vaciar B1,Verter B2- B1,Vaciar B1,Verter B2- B1,Llenar B2,Verter B2- B1]

Estados camino:
[estado(0,0),estado(5,0),estado(5,8),estado(0,8),estado(5,3),estado(0,3),estado(3,0),estado(3,8),estado(5,6),estado(0,6),estado(5,1),estado(0,1),estado(1,0),estado(1,8),estado(5,4)]
true