

Práctica No. 2

Sistemas de producción en Common Lisp

Objetivo

Implementar sistemas de producción en Common Lisp.

Introducción

Cuando un problema se representa como un espacio de estados, la solución del mismo se vuelve la búsqueda de un estado meta partiendo de un estado inicio. El espacio de estados puede estar representados explícitamente o implícitamente. Cuando la representación es implícita, un sistema de producción es una herramienta común para obtener los estados siguientes de un estado dado.

Actividades

1.- Sistema de Producción para el problema de las Jarras

1.1.- Captura el sistema de producción siguiente, que se utiliza para la solución del problema de las 2 jarras.

```
;;;-----  
;;; spjarras.lisp  
;;; Sistema de produccion para el problema de las 2 jarras  
;;; Estado: (A,B), donde A = Jarra de 4 litros, B = Jarra de 3 litros  
;;;-----  
(defun jarra-a (estado) (car estado))  
(defun jarra-b (estado) (cadr estado))  
(defun haz-estado (p s) (list p s))  
  
(defun estados-nuevos (estado)  
  (remove-if #'(lambda (x) (endp x))  
    (list  
      (llena-a estado)  
      (llena-b estado)  
      (vacía-a-en-b estado)  
      (vacía-b-en-a estado)  
      (vacía-a estado)  
      (vacía-b estado)  
    )  
  )  
)  
  
(defun llena-a (estado)  
  (cond ((< (jarra-a estado) 4) (haz-estado 4 (jarra-b estado))))  
)  
  
(defun llena-b (estado)  
  (cond ((< (jarra-b estado) 3) (haz-estado (jarra-a estado) 3)))  
)  
  
(defun vacía-a-en-b (estado)  
  (let ( (p (jarra-a estado))  
        (s (jarra-b estado)))  
    (cond ((zerop p) nil) ; Jarra A vacía no puede vaciarse  
          ((= s 3) nil) ; Jarra B llena no puede llenarse  
          ((<= (+ p s) 3) ; Vacía A en B  
            (haz-estado 0 (+ p s)))  
          (t (haz-estado (- (+ p s) 3) 3)))  
    )  
  )  
)  
  
(defun vacía-b-en-a (estado)  
  (let ( (p (jarra-a estado))  
        (s (jarra-b estado)))
```

```

      (cond ((zerop s) nil) ; Jarra B vacia no puede vaciarse
            ((= p 4) nil) ; Jarra A llena no puede llenarse
            ((<= (+ p s) 4) ; Vacía B en A
             (haz-estado (+ p s) 0))
            (t (haz-estado 4 (- (+ p s) 4))))
    )
)

(defun vacia-a (estado)
  (cond ((> (jarra-a estado) 0) (haz-estado 0 (jarra-b estado))))
)

(defun vacia-b (estado)
  (cond ((> (jarra-b estado) 0) (haz-estado (jarra-a estado) 0)))
)

```

De esta manera, para obtener los sucesores de un estado, desde el intérprete de Common Lisp tecleamos: (estados-nuevos '(a b)). Por ejemplo: Para obtener los sucesores del estado (1 1), tecleamos: (estados-nuevos '(1 1))

1.2.- Ejecuta el programa y reporta los sucesores que obtiene para los siguientes estados:

Estado (0 0), Sucesores: _____
 Estado (4 0), Sucesores: _____
 Estado (1 3), Sucesores: _____
 Estado (4 3), Sucesores: _____
 Estado (3 3), Sucesores: _____
 Estado (0 3), Sucesores: _____
 Estado (4 2), Sucesores: _____
 Estado (0 2), Sucesores: _____

2.- Sistema de producción para el problema de los misioneros y canibales

2.1.- Para el problema de los misioneros y canibales, vamos a definir cada estado como una triada: (PB Mizq Cizq) Donde: PB: Indica la posición de la barca: {izq,der}, Mizq: Número de misioneros en el margen izquierdo del río: {0, 1, 2, 3} y Cizq: Número de canibales en el margen izquierdo del río: {0, 1, 2, 3}.

De esta forma, el estado inicial es (Izq 3 3) y el estado final es (Der 0 0)

Ademas tenemos que Mder=3-Mizq y Cder=3-Cizq

Una propuestas de reglas para el sistema de producción de este problema son las siguientes:

- 1.- Si ((PB=izq) AND (Mizq >= 2) AND (Mizq-2 >= Cizq))
 ENTONCES Nuevo-estado = (Der Mizq-2 Cizq) ; Mueve 2 misioneros al margen derecho
- 2.- Si ((PB=izq) AND (Cizq >= 2))
 ENTONCES Nuevo-estado = (Der Mizq Cizq-2) ; Mueve 2 canibales al margen derecho
- 3.- Si ((PB=izq) AND (Mizq >= 1) AND (Cizq >= 1))
 ENTONCES Nuevo-estado = (Der Mizq-1 Cizq-1) ; Mueve 1 misionero y 1 canibal al margen derecho
- 4.- Si ((PB=izq) AND (Mizq >= 1) AND (Mizq-1 >= Cizq))
 ENTONCES Nuevo-estado = (Der Mizq-1 Cizq) ; Mueve 1 misionero al margen derecho
- 5.- Si ((PB=izq) AND (Cizq >= 1) AND (Mizq >= Cizq-1))
 ENTONCES Nuevo-estado = (Der Mizq Cizq-1) ; Mueve 1 canibal al margen derecho
- 6.- Si ((PB=der) AND (Mder >= 2) AND (Mder-2 >= Cder))
 ENTONCES Nuevo-estado = (Izq Mizq+2 Cizq) ; Mueve 2 misioneros al margen izquierdo
- 7.- Si ((PB=der) AND (Cder >= 2))
 ENTONCES Nuevo-estado = (Izq Mizq Cizq+2) ; Mueve 2 canibales al margen izquierdo
- 8.- Si ((PB=der) AND (Mder >= 1) AND (Cder >= 1) AND (Mder-1 >= Cder-1))
 ENTONCES Nuevo-estado = (Izq Mizq+1 Cizq+1) ; Mueve 1 misionero y 1 canibal al margen izquierdo

