Práctica 5- Problemas granjero y misioneros

1. Realizar un programa en LISP que compruebe si una lista de acciones (operadores) resuelve o no el problema del granjero visto en clase de teoría.

Problema:

Un granjero está con un lobo, una cabra y una col en una orilla del río. Desea pasar a la otra orilla. Dispone de una barca en la que sólo puede llevar una cosa cada vez o se hunde. El lobo se come a la cabra si no está el granjero. La cabra se come la col si no está el granjero.

Guía para resolver el problema:

1. Construir los estados.
2. Diseñar funciones de acceso a las componentes de cada estado.
3. Declarar el estado inicial.
4. Declarar el estado final o la función de comprobación de si es estado final. ¿Se necesitan funciones auxiliares?
5. Diseñar la función ES-SEGURO, que, dado un estado, compruebe si es una posición válida o no.
6. Declarar la lista de operadores.
7. Diseñar las funciones de cada elemento de la lista de operadores.
8. Reutiliza de la práctica anterior las funciones APLICA y VERIFICA.

2. Realizar un programa en LISP que compruebe si una lista de acciones (operadores) resuelve o no el problema de los misioneros con 3 misioneros, 3 caníbales y una barca de 2 personas de capacidad.

Problema:

En la orilla izquierda de un río se encuentran M misioneros y C caníbales (M>=C) y una barca en la que caben S personas. El problema consiste en determinar cómo pueden trasladarse los misioneros y los caníbales a la orilla derecha teniendo en cuenta que:

- En la orilla en la que no está la barca hay misioneros y su número no puede ser menor que la de caníbales en dicha orilla.
- La barca no puede viajar vacía.

Guía para resolver el problema:

1. Declaración de variables globales:
   - (defvar *misioneros* 3)
   - (defvar *canibales* 3)
   - (defvar *capacidad* 2)
2. Construir los estados.
3. Diseñar funciones de acceso a las componentes de cada estado.
4. Declarar el estado inicial.
5. Declarar el estado final o la función de comprobación de si es estado final. ¿Se necesitan funciones auxiliares?
6. Diseñar la función ES-SEGURO, que, dado un estado, compruebe si es una posición válida o no.
7. Declarar la lista de operadores.
8. Diseñar las funciones de cada elemento de la lista de operadores.
9. Reutiliza de la práctica anterior las funciones APLICA y VERIFICA.