Robots Móviles

Practica 4
Planeación de Acciones Usando un Sistema Basado en Reglas

Objetivo: Familiarizar al alumno con la planeacion de acciones usando sistemas basados en reglas.

Desarrollo: Para cada uno de los siguientes apartados, realizar los programas que se piden.

Duración: Tres semanas

1.- El apéndice A contiene el código de un sistema que controla la manipulación de cubos en diferentes configuraciones, usando el lenguaje basado en reglas CLIPS. Pruebe este código y entienda su funcionamiento.

2.- En el apéndice B, figura 1, en el cuarto denominado “Depósito” se encuentran seis objetos los cuales serán relocalizados de acuerdo a la figura 2. Para poder mover los objetos y colocarlos en otro lugar éstos deberán ser movidos siguiendo un orden, es decir que para mover al bloque A se necesita mover primero los bloques C y B a un lugar libre. Para colocarlos después en su posición final también se necesita hacerlo siguiendo un orden preestablecido.

Configure un sistema de movimiento de objetos de un lugar a otro usando CLIPS, teniendo una representación del mundo con hechos, reglas (axiomas) y operadores.

Este sistema deberá ser flexible con respecto a solucionar cualquier configuración inicial y final, la salida de éste será un archivo utilizando los siguientes operadores:

goto room
goto x,y,tetha
find configuration_objects
find object
grasp object
release object

3.- Pruebe su sistema utilizando primero el simulador utilizado en las practicas y después en el TurtleBot, leyendo el archivo generado en el punto anterior y ejecutando los comandos. Para la planeación de movimientos del robot entre cuartos use una red topológica y algoritmos de busqueda, tales como el algoritmo de Dijkstra, A*, etc. Para obstaculos desconocidos utilice cualquiera de las siguientes técnicas: algoritmos de máquinas de estados o campos potenciales.

APENDICE A
Mundo de los Bloques

```
;;;=======================================================
;;;   Programa del Mundo de los Bloques
;;;   Para ejecutarlo solamente carguelo, de reset y ejecutelo
;;;=======================================================

(deftemplate goal (slot move) (slot on-top-of))
(deffacts initial-state
   (stack A B C)
   (stack D E F)
   (goal (move C) (on-top-of E))
   (stack))

(defrule move-directly
   ?goal <- (goal (move ?block1) (on-top-of ?block2))
   ?stack-1 <- (stack ?block1 $?rest1)
   ?stack-2 <- (stack ?block2 $?rest2)
   =>
   (retract ?goal ?stack-1 ?stack-2)
   (assert (stack $?rest1))
   (assert (stack ?block1 ?block2 $?rest2))
   (printout t ?block1 " moved on top of "
             ?block2 "." crlf))

(defrule move-to-floor
   ?goal <- (goal (move ?block1) (on-top-of floor))
   ?stack-1 <- (stack ?block1 $?rest)
   =>
   (retract ?goal ?stack-1)
   (assert (stack ?block1))
   (assert (stack $?rest))
   (printout t ?block1 " moved on top of floor." crlf))

(defrule clear-upper-block
   (goal (move ?block1))
   (stack ?top $? ?block1 $?)
   =>
   (assert (goal (move ?top) (on-top-of floor))))

(defrule clear-lower-block
   (goal (on-top-of ?block1))
   (stack ?top $? ?block1 $?)
   =>
   (assert (goal (move ?top) (on-top-of floor))))
```

APENDICE B
Mundo de los Bloques

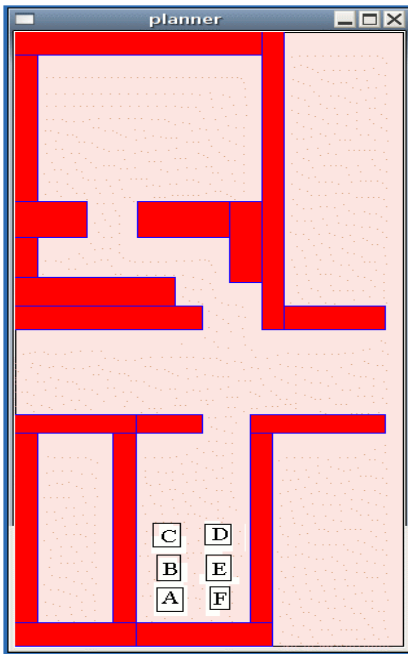

Figura 1.

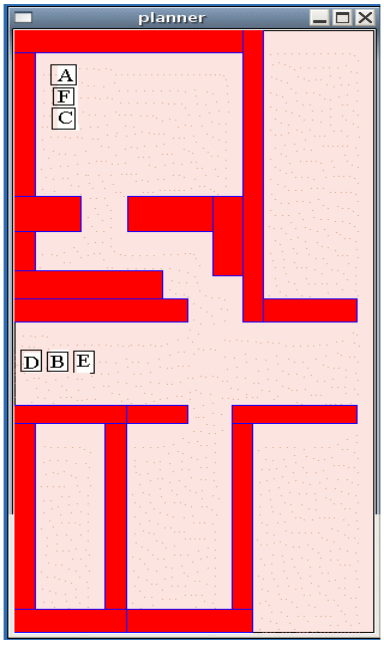

Figura 2.