

Como lo que se hizo fue un agente lógico basado en conocimiento, lo primero que se hizo fue pensar en el problema general y problemas particulares a resolver, y así determinar que conocimiento necesitaría el agente para resolver el problema, como lo adquiriría (hacer la base de conocimiento) y lo usaría inteligentemente para llegar al oro y sacarlo sin morir en el intento. Las tareas que se pensaron se necesitarían hacer fueron: ¿Cómo el agente adquiriría el conocimiento?, ¿Cómo sabría que llegó ya al oro?, ¿Cómo se regresaría a la posición inicial?, ¿Cómo sabría si es seguro o no avanzar a un lugar?.

Las reglas del juego son muy sencillas, si hay brisa en una de las casillas alrededor del agente hay un hoyo, si hay mal olor alrededor del agente está el Humpus, y si hay brillo, está el oro; y teniendo en cuenta todo esto, se crea la base de conocimiento del agente, y así usarla para resolver el problema.

Como prolog es un lenguaje donde únicamente se le pueden dar hechos y reglas, para que el agente pudiera determinar las dimensiones del mapa, se le pasó un hecho, el cual se adquiriría a su base de conocimiento, donde pudiera determinar los límites del mapa.

mapa([_,1], sur).

mapa([_,4], norte).

mapa([1,_], oeste).

mapa([4,_], este).

Una vez que el agente identifica cuáles son sus limitaciones se hace la recopilación de la información relevante, o sea añadir a la base de conocimiento las ubicaciones donde hay mal olor o donde hay brisa, tal y como se muestra a continuación:

brisa([2,1]). olor([1,2]).

brisa([3,2]). olor([1,4]).

brisa([4,1]). olor([2,3]).

brisa([2,3]).

brisa([3,4]).

Brisa([4,3]).

Y con esto el agente lógico ya puede traducir los conceptos importantes.

Eso quiere decir que en mi implementación, el agente ya sabe las ubicaciones de los malos olores y de las brisas, pero realmente no sabe donde está el oro, el wumpus o los hoyos. La base de conocimiento que se va adquiriendo básicamente es el recorrido que ha realizado el agente para que pueda regresar por el mismo camino sano y salvo hasta la posición de inicio.

La posición inicial del agente se define en la posición (1,1) mirando al norte.

Ya que se define la base de conocimiento y las tareas a realizar, se inicia la implementación con las reglas que el agente debe seguir, o los axiomas a tener en cuenta.

Primero mover al agente, donde los movimientos posibles que tiene el agente se definieron como:

1)avanzar, 2)mover derecha, 3)mover izquierda.

Cada que avanza, se actualiza la base de conocimiento para saber los lugares por los que ha pasado, guardándolos en una lista.

Lo primero que hace el agente antes de avanzar es verificar si en la posición que está, ahí está el oro, en caso de que no, el agente hace un proceso lógico para saber si avanzar al frente es seguro o no, de la siguiente manera:

supongamos que el agente quiere avanzar a la posición $[x,y]$.

Para saber que no hay wumpus ahí se hace lo siguiente:

$(\text{si en } [x+1,y] \text{ NO hay mal olor } \vee \text{ si en } [x,y+1] \text{ No hay mal olor } \vee \text{ si en } [x-1,y] \text{ No hay mal olor } \vee \text{ si en } [x,y-1] \text{ NO hay mal olor}) \Leftrightarrow \text{NO hay Humpus en } [x,y]$.

lo mismo para evitar los hoyos:

$(\text{si en } [x+1,y] \text{ NO hay brisa } \vee \text{ si en } [x,y+1] \text{ No hay brisa } \vee \text{ si en } [x-1,y] \text{ No hay brisa } \vee \text{ si en } [x,y-1] \text{ NO hay brisa}) \Leftrightarrow \text{NO hay hoyo en } [x,y]$.

es seguro si:

$\text{NO hay Humpus en } [x,y] \wedge \text{NO hay hoyo en } [x,y]$.

Si es seguro el agente avanza y vuelve a verificar si ya encontró el oro.

De ésta manera sigue avanzando el agente hasta encontrar el oro y regresar por el camino que recorrió pero ahora hacía atrás.

En caso de que no haya manera solución, el agente lo sabrá también y la salida sería que no hay solución.