**RESOLVER PROBLEMAS MEDIANTE BÚSQUEDA**

**LISETH VALERIA QUIÑONES DEL CASTILLO**

**UNIDAD CENTRAL DEL VALLE DEL CAUCA**

**FACULTAD DE INGENIERIA**

**INTELIGENCIA ARTIFICIAL**

**TULUA 2019**

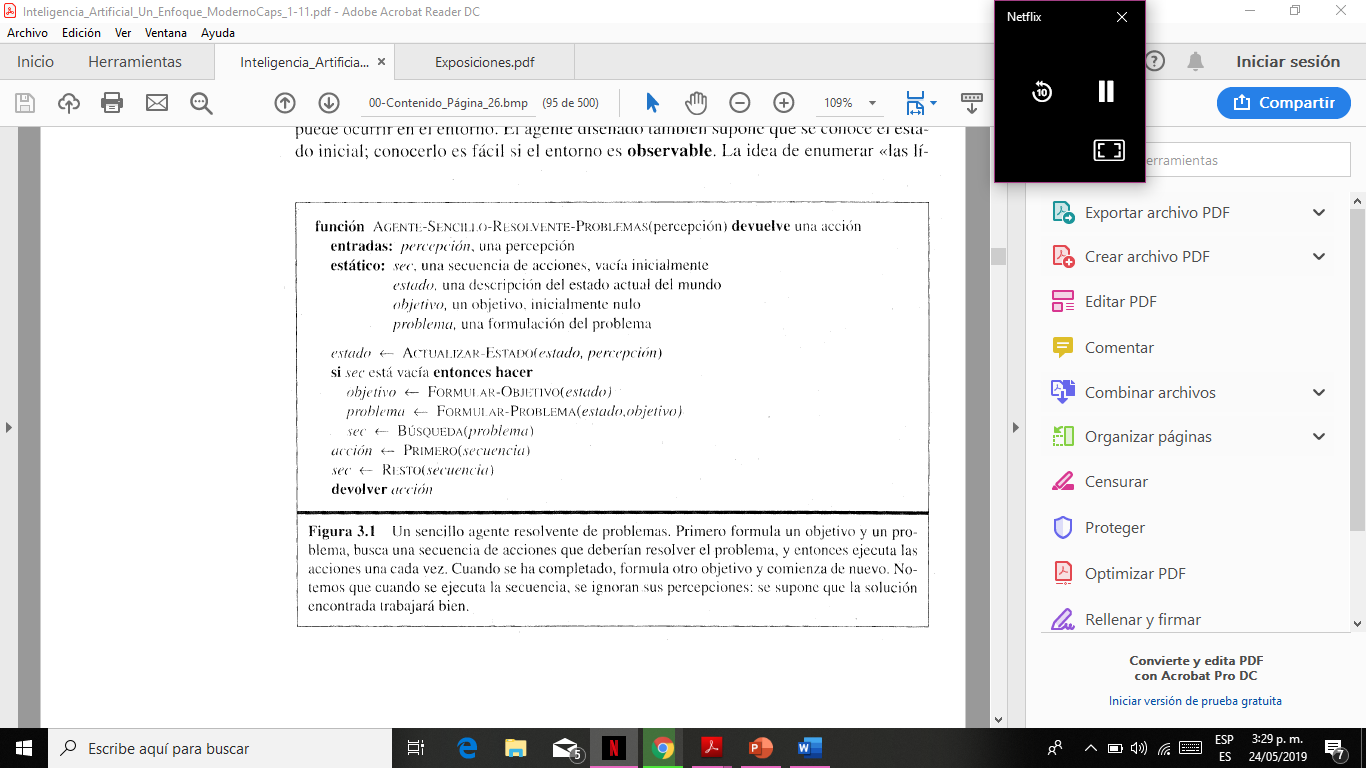
INTRODUCCIÓN 

En esta ocasión haremos una revisión de otro tipo de agente como el agente resolvente problemas. Los agentes resolventes-problemas deciden qué hacer para encontrar secuencias de acciones que conduzcan a los estados deseables y conoceremos los diferentes algoritmos de búsqueda informada, que tienen cierta idea de dónde buscar las soluciones.

AGENTES RESOLVENTES-PROBLEMAS

Los agentes resolventes–problemas son agentes basados en objetivo, el cual tiene como función principal decidir ¿Qué hacer? Para encontrar las secuencia de acciones que lo conlleve a los estados deseables. Estos agentes están constituido por os elementos que son: una solución y un problema. Estos agentes deben seguir un objetivo, el cual es el encargado de organizar el comportamiento del agente, según el problema:

El primer paso para solucionar un problema es la formulación de objetivo basado en la situación actual y la medida de rendimiento del agente. Es el proceso de decidir qué acciones y estados tenemos que considerar. La fase de búsqueda, en esta se analizan todas las secuencias posibles de acciones que nos conlleven a los estados de valores conocidos, para así escoger la mejor secuencia de acciones, luego de esto se ejecutan la secuencia de acciones que fue elegida en la fase de búsqueda.



Problemas y soluciones bien definidos

Un problema consta de cuatro partes:

Estado Inicial: es la situación inicial donde comienza el agente. Ejemplo si una agente está en Ecuador y quiere ir a Francia, el estado inicial es el agente esta en Ecuador.

Test Objetivo: determina si un estado es un estado objetivo. Muchas veces es un conjunto especifico de posibles estados objetivos y el test objetivo lo que hace es comprobar si el estado pertenece a ellos. Ejemplo si una agente está en Ecuador y quiere ir a Francia, el test objetivo es llegar a Francia.

Función Sucesor: son las posibles acciones que un agente realiza y con dichas acciones se obtiene como resultado un estado. Es decir que una función sucesor es un estado resultante de una acción. Ejemplo si una agente está en Ecuador y quiere ir a Francia, la función sucesor sería ir a Francia, estar en Francia.

Función Costo: el cual asigna un costo numérico a un camino. Con la función costo se puede reflejar las medidas de rendimiento de un agente resolventes-problemas.

Los algoritmos de búsqueda se juzgan sobre la base de completitud, optimización, complejidad en tiempo, complejidad en espacio.

Formular los problemas

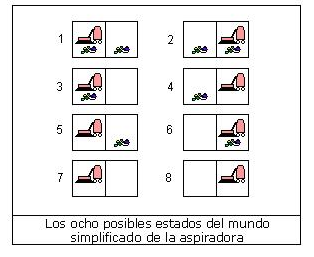
Al proceso de eliminar detalles de una representación se le llama abstracción.

La abstracción es válida si podemos ampliar cualquier solución abstracta a una solución en el mundo mas detallada es decir que es posible utilizarla si al realizar cada una de las soluciones es más fácil que en la original. Si no fuera por la capacidad de construir abstracciones útiles, los agentes inteligentes quedarían totalmente absorbidos por el mundo real.

EJEMPLOS DE PROBLEMAS

El mundo de la aspiradora.

En este mundo hay dos posibles ubicaciones en ellas puede o no puede haber mugre y el agente se encuentra en una de las dos.



El mundo puede asumir 8 posibles estados. Son tres las acciones que el agente puede emprender en el mundo de la aspiradora. La meta es eliminar toda la mugre.

Problemas bien definidos y soluciones

• Problema: Un problema en realidad es un conjunto de información que el agente utiliza para decidir lo que va a hacer.

• Estado inicial: Es donde el agente sabe que allí es en el que se encuentra.

• Operador: Denota la descripción de una acción que nos servirá para alcanzar un estado.

• Espacio de estado: Es el conjunto de todos los estados que pueden alcanzarse a partir del estado inicial mediante cualquier secuencia de acciones.

• Ruta: Es cualquier secuencia de acciones que me permiten pasar de un estado a otro.

• Prueba de meta.- Se aplica a la descripción de un solo estado para saber si se trata de un estado meta.

• Costo de ruta.- Es una función mediante la cual se asigna un costo a un ruta determinada.

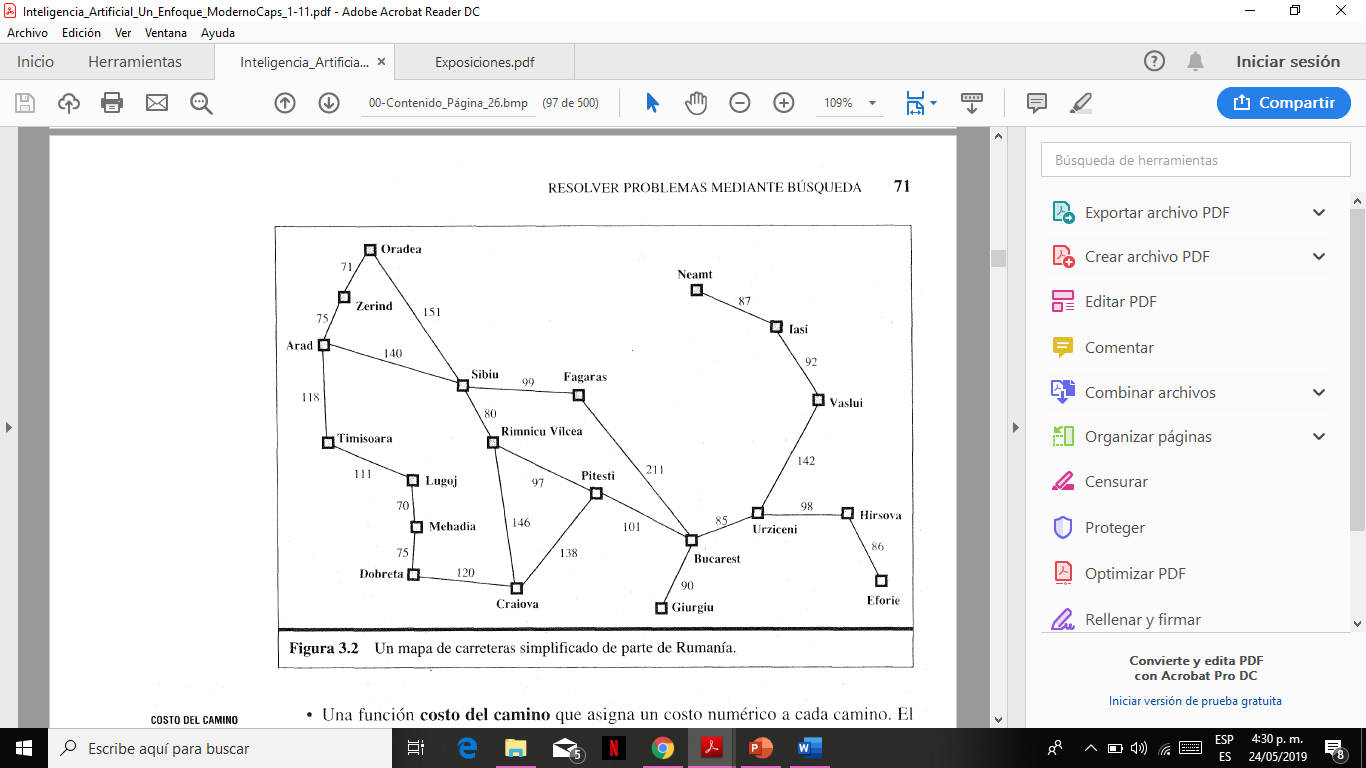
• Solución.- Es la salida producida por un algoritmo de búsqueda, es decir, una ruta que va del estado inicial al estado que satisface la prueba de meta.

• Espacio de conjunto de estado.-Si mediante un operador relaciono un conjunto de estados según los resultados obtenidos, yo estoy construyendo un espacio denominado espacio de conjunto de estados.



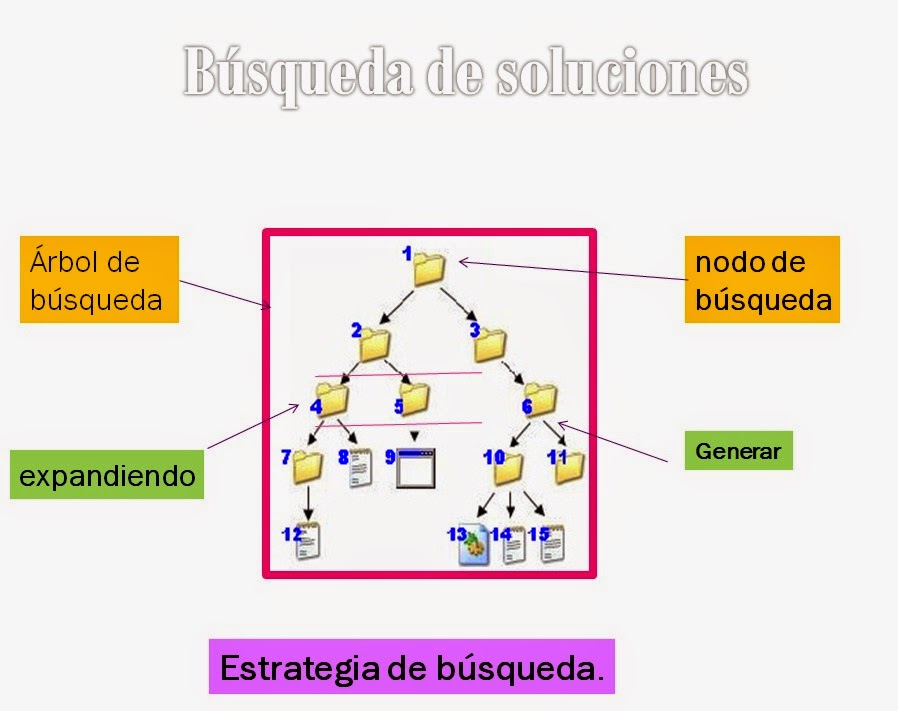
El espacio del conjunto de espacios del mundo de la aspiradora simplificado, sin sensores. Dentro de las cajas de línea punteada hay un conjunto de estados. En un punto determinado, el agente está dentro de un conjunto de estados, pero ignora en cuál. El conjunto del estado inicial (total ignorancia) está en la caja superior del centro. Las acciones se representan por arcos identificados. Por claridad se han omitido los auto bucles.

Los problemas turísticos estan estrechamcnte relacionaclos con los problemas de busqueda de una ruta pero con una importante diferencia. Consideremos la siguiente figura3.2 al menos una vez comenzando y finalizando en Bucarest como en la busqueda de rutas las acciones corresponden a viajes entre ciudades adyacentes, el espacio de estados es sin embargo absolutamente diferentes. Cada estado debe incluir no solo la localizacion actual sino tambien las ciudades que el agente ha visitado . el estado inicial seria bucarest. Un estado termino medio tipico seria en Vasliu y el test comprobaria si el agente esta en bucarest y ha visitado las 20 ciudades.



BUSQUEDA DE SOLUCIONES

Buena parte de los esfuerzos invertidos en el área de la búsqueda han quedado en la determinación de la estrategia de búsqueda adecuada para un problema.



Árbol de búsqueda: generado por el estado inicial y la función sucesor, definiendo así el espacio de estados. En general, podemos tener un grafo de búsqueda mas que un árbol, cuando el mismo estado puede alcanzarse desde varios caminos.

La raíz del árbol de búsqueda es el nodo de búsqueda que corresponde al estado inicial El primer paso es comprobar si este es un estado objetivo. Como no estamos en un estado objetivo, tenemos que considerar otros estados. Esto se hace expandiendo el estado actual; es decir aplicando la función sucesor al estado actual y generar así un nuevo conjunto de estados. El estado a expandir está determinado por la estrategia de búsqueda.

ESTRATEGIAS DE BÚSQUEDA NO INFORMADA

Esta sección trata cinco estrategias de búsqueda englobadas bajo el nombre de búsqueda no informada (llamada también búsqueda a ciegas). El término significa que ellas no tienen información adicional acerca de los estados más allá de la que proporciona la definición del problema. Todo lo que ellas pueden hacer es generar los sucesores y distinguir entre un estado objetivo de uno que no lo es. Las estrategias que saben si un estado no objetivo es más prometedor que otro se llaman búsqueda informada o búsqueda heucarística.

Búsqueda primero en anchura

La búsqueda primero en anchura es una estrategia sencilla en la que se expande primero el nodo raíz, a continuación se expanden todos los sucesores del nodo raíz, después sus sucesores, etc. En general, se expanden todos los nodos a una profundidad en el árbol de búsqueda antes de expandir cualquier nodo del próximo nivel. Es completo, óptimo para costos unidad, y tiene la complejidad en tiempo y en espacio de La complejidad en espacio lo hace poco práctico en la mayor parte de casos.

Búsqueda de costo uniforme

La búsqueda primero en anchura es óptima cuando todos los costos son iguales, porque siempre expande el nodo no expandido más superficial. Con una extensión sencilla, podemos encontrar un algoritmo que es óptimo con cualquier función costo. En vez de expandir el nodo más superficial, la búsqueda de costo uniforme expande el nodo n con el camino de costo más pequeño. Notemos que si todos los costos son iguales, es idéntico a la búsqueda primero en anchura. La búsqueda de costo uniforme no se preocupa por el número de pasos que tiene un camino, pero sí sobre su coste total.

Búsqueda primero en profundidad

La búsqueda primero profundidad siempre expande el nodo más profundo en la frontera actual del árbol de búsqueda. La búsqueda procede inmediatamente al nivel más profundo del árbol de búsqueda, donde los nodos no tienen ningún sucesor. Cuando esos nodos se expanden, son quitados de la frontera, así entonces la búsqueda «retrocede» al siguiente nodo más superficial que todavía tenga sucesores inexplorados.

Búsqueda de profundidad limitada

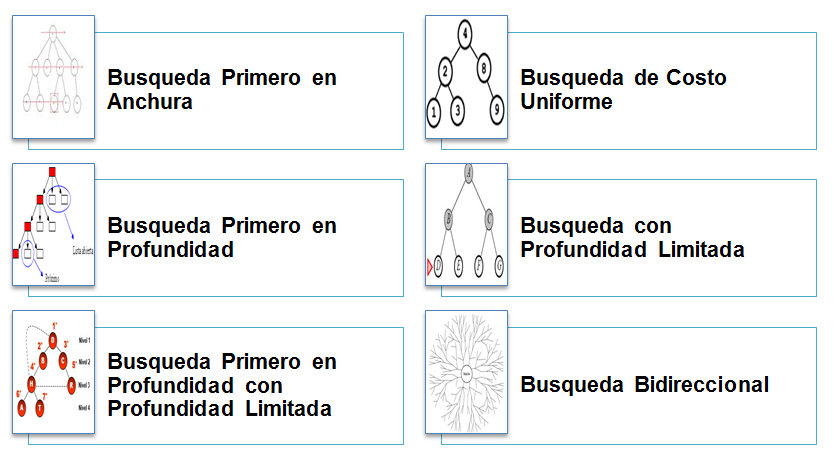
Impone un límite de profundidad fijo a una búsqueda primero en profundidad. Se puede aliviar el problema de árboles ilimitados aplicando la búsqueda primero en profundidad con un límite de profundidad t predeterminado. Es decir, los nodos a profundidad i se tratan como si no tuvieran ningún sucesor. A esta aproximación se le llama búsqueda de profundidad imitada El límite de profundidad resuelve el problema del camino infinito.

Búsqueda primero en profundidad con profundidad iterativa

La búsqueda con profundidad iterativa (o búsqueda primero en profundidad con profundidad iterativa) es una estrategia general, usada a menudo en combinación con la búsqueda primero en profundidad, la cual encuentra el mejor límite de profundidad. Esto se hace aumentando gradualmente el límite (primero 0, después 1, después 2, etcétera) hasta que encontramos un objetivo. Esto ocurrirá cuando el límite de profundidad alcanza d, profundidad del nodo objetivo.

Búsqueda bidireccional

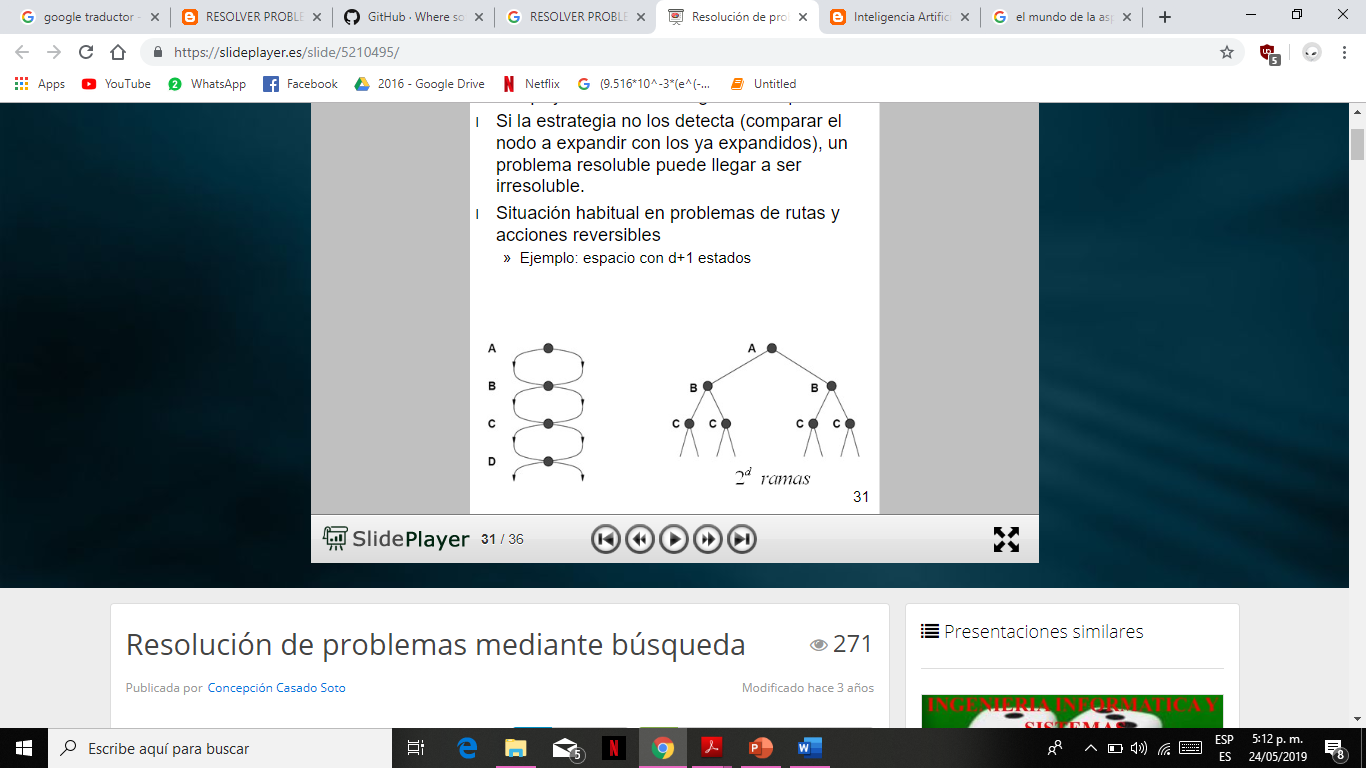
La idea de la búsqueda bidireccional es ejecutar dos búsquedas simultáneas: una hacia delante desde el estado inicial y la otra hacia atrás desde el objetivo, parando cuando las dos búsquedas se encuentren en el centro. La búsqueda bidireccional se implementa teniendo una o dos búsquedas que comprueban antes de ser expandido si cada nodo está en la frontera del otro árbol de búsqueda; si esto ocurre, se ha encontrado una solución. Puede reducir enormemente la complejidad en tiempo, pero no es siempre aplicable y puede requerir demasiado espacio. El algoritmo de Búsqueda-Grafos elimina todos los estados duplicados.



EVITAR ESTADOS REPETIDOS

* La repetición de estados incrementa la complejidad de la estrategia de búsqueda.
* Si la estrategia no los detecta (comparar el nodo a expandir con los ya expandidos), un problema resoluble puede llegar a ser irresoluble.
* Situación habitual en problemas de rutas y acciones reversibles.

Ejemplo: espacio con d+1 estados



Para los d+1 estados (d es la profundidad máxima). El árbol de búsqueda contendrá 2d ramas. Poda.

Para evitar que se repitan estados, se pueden considerar tres métodos:

* No generar un nodo hijo de un nodo si los dos pertenecen al mismo estado
* Evitar ramas con ciclos (en un camino desde el nodo inicial, hay dos nodos que pertenecen el mismo estado) El método 2) incluye al 1)
* Si al generar un nodo, su estado asociado, ya ha sido generado por otro nodo, eliminar el nodo peor (y sus descendientes) del árbol de búsqueda. El método 3) incluye al 2) y, por tanto, al 1). Este método es el más caro (hay que mantener todos los nodos en memoria).

Estructuras de datos

* Listas cerradas (nodos expandidos)
* Listas abiertas (frontera de nodos no expandidos)

BUSQUEDA CON INFORMACION PARCIAL

El agente puede calcular exactamente cuál es el estado resultado de cualquier secuencia de acciones y siempre sabe en qué estado está. Su percepción no proporciona ninguna nueva información después de cada acción. ¿Qué pasa cuando el conocimiento de los estados o acciones es incompleto? Encontramos que diversos tipos de incompleto conducen a tres tipos de problemas distintos:

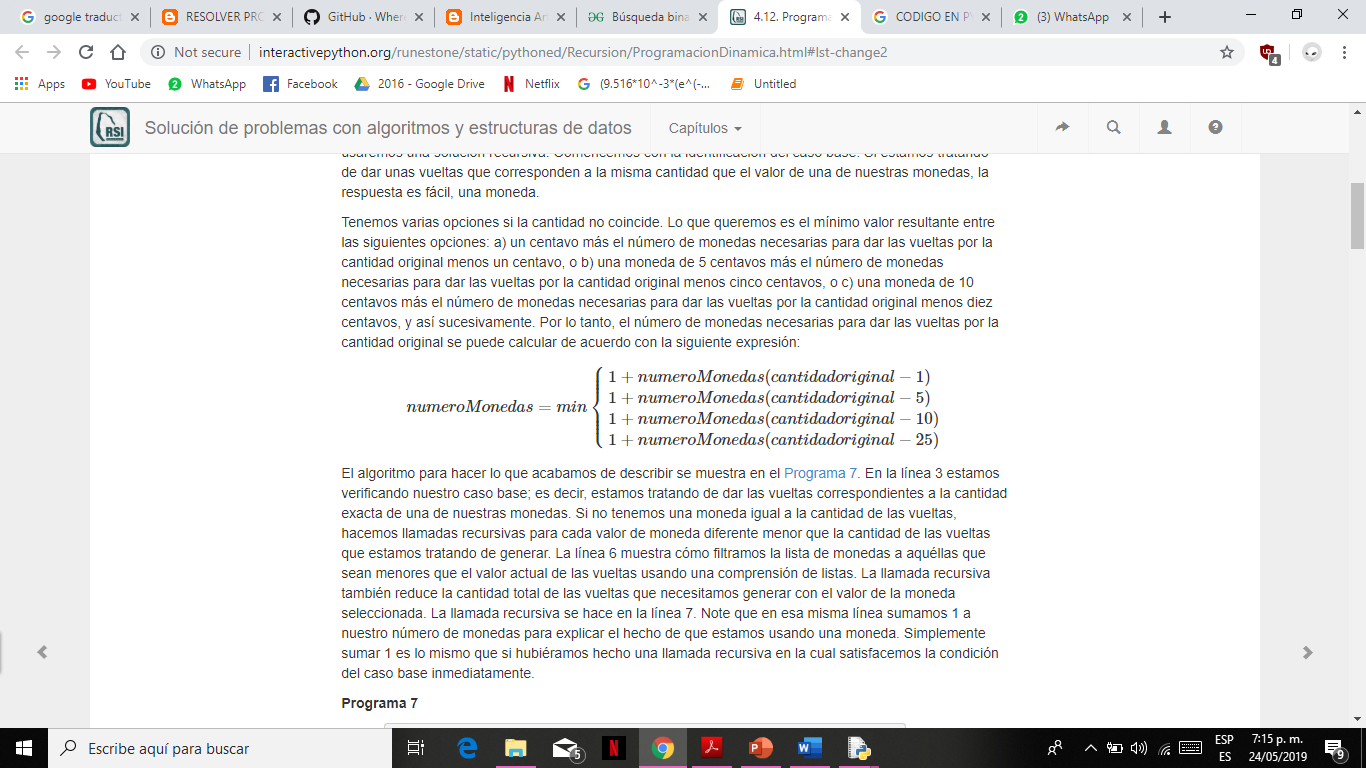
1. Problemas sin sensores (también llamados problemas conformados): si el agente no tiene ningún sensor, entonces (por lo que sabe) podría estar en uno de los posibles estados iniciales, y cada acción por lo tanto podría conducir a uno de los posibles estados sucesores.

2. Problemas de contingencia: si el entorno es parcialmente observable o si las acciones son inciertas, entonces las percepciones del agente proporcionan nueva- información después de cada acción. Cada percepción posible define una contingencia que debe de planearse. A un problema se le llama entre adversarios si la incertidumbre está causada por las acciones de otro agente.

3. Problemas de exploración: cuando se desconocen los estados y las acciones del entorno, el agente debe actuar para descubrirlos. Los problemas de exploración pueden verse como un caso extremo de problemas de contingencia.

EJERCICOS :

1. Entre las siguientes opciones determinar la cantidad de vueltas de moneda para 63 centavos a partir de estos datos : a) un centavo más el número de monedas necesarias para dar las vueltas por la cantidad original menos un centavo, o b) una moneda de 5 centavos más el número de monedas necesarias para dar las vueltas por la cantidad original menos cinco centavos, o c) una moneda de 10 centavos más el número de monedas necesarias para dar las vueltas por la cantidad original menos diez centavos, y así sucesivamente. Por lo tanto, el número de monedas necesarias para dar las vueltas por la cantidad original se puede calcular de acuerdo con la siguiente expresión:



1. Es un algoritmo de programación dinámica para solucionar nuestro problema de dar las vueltas. vueltasProgDin tiene tres parámetros: una lista de valores válidos de monedas, la cantidad de vueltas que queremos completar y una lista del número mínimo de monedas necesarias para completar cada valor. Cuando la función termine, minMonedas contendrá la solución para todos los valores desde 0 hasta el valor de vueltas.

En este ejercicio se debe usar la lista del anterior ejercicio y recolectar los mismos 63 centavos.

CONCLUSIÓN

El estudio de los agentes resolventes-problemas es necesario dentro de la Inteligencia Artificial, ya que estos agentes son aquellos que buscan darle soluciones a problemas que son frecuentes en la vida cotidiana. Teniendo componentes que son indispensable conocer sus conceptos para poder resolver problemáticas. Los agentes tienen como prioridad algoritmos que hagan que el agente sea capaz de darle soluciones óptimas a los problemas, es por esto que en la inteligencia artificial ha desarrollado mecanismo eficaces para la resolución de problemas.