UNIVERSITAT DE BARCELONA

Departament de Matemàtica Aplicada i Anàlisi
Gran Via de les Corts Catalanes, 585
08007 Barcelona
Tel . 93 402 16 50, Fax . 93 403 16 01

Examen de Inteligencia Artificial: primer parcial curso 2012

DNI estudiante:
Nombre y apellidos alumno que lo corrige:

Preguntas de teoría

Contestar si es Verdadero (V) o Falso (F) cada apartado de cada pregunta y en caso de que sea falso, decir el por qué. Cada pregunta vale 1 pt, cada uno de los 4 subapartados vale 0.25 pts. Aciertos como V cuentan como 0.25, aciertos como F cuentan como 0.1 si no/mal justificado, o como 0.25 si bien razonado. Los errores no descuentan. Cualquier otro valor no cuenta.

Ejemplo: Este examen:

a. Se hace por primera vez en esta asignatura: V -> puntuación: 0.25
b. Va a durar 3 horas: V -> la puntuación es -0.1
c. Va a durar 2 horas: F -> puntuación: 0.1
d. Va a durar 2.5 horas F: durará media hora -> puntuación 0.25

1. La inteligencia artificial:
   a. Es un área reciente (oficialmente desde 1996) (F: 1956)
   b. Es un área que se creó oficialmente en 1956 (T)
   c. Es reciente en comparación con el estudio de la inteligencia, que se hace desde muy antiguo en disciplinas como la filosofía. (T)
   d. Se puede ver como ciencia pero no como ingeniería (F: también puede ser ingeniería que construye máquinas inteligentes)

2. La definición de inteligencia Artificial:
   a. "La Inteligencia computacional es el estudio del diseño de agentes inteligentes" corresponde a la taxonomía de sistemas que piensan racionalmente: (F: son los que actúan racionalmente)
   b. "El nuevo y excitante esfuerzo de hacer que los computadores piensen... máquinas con mentes, en el más amplio sentido literal" corresponde a la taxonomía de sistemas que piensan como humanos (V)
   c. Que dice que se trata de hacer sistemas que actúen como humanos sigue el enfoque de la "prueba de Turing" (V)
   d. Que dice que se trata de hacer sistemas que piensan racionalmente sigue el enfoque cognoscitivo (F: es el lógico)

3. La racionalidad está relacionada con:
   a. La maximización de los objetivos establecidos (V)

UNIVERSITAT DE BARCELONA

Departament de Matemàtica Aplicada i Anàlisi
Gran Via de les Corts Catalanes, 585
08007 Barcelona
Tel . 93 402 16 50, Fax . 93 403 16 01

b. Los procesos mentales que los generan (F: es independiente)

c. La maximización de la utilidad esperada (V)

d. La optimización en términos absolutos de las decisiones (F: no se pretende garantizar la optimalidad)

4. Los agentes que resuelven problemas:

   a. Son reactivos (F: son deliberativos)

   b. Se basan en objetivos para buscar las secuencias de acciones que los consigan (V)

   c. Buscan de forma no dirigida entre las posibles acciones a realizar (F: es dirigida por los objetivos)

   d. Necesitan que se les formule el problema a resolver (V)

5. Para resolver problemas:

   a. Se definen las 4 etapas de (1) formulación de objetivos, (2) formulación de acciones posibles, (3) Búsqueda y (4) Ejecución. (F: el 2 es la formulación del problema, que incluye acciones y estados)

   b. Es necesario ejecutar acciones mientras se van buscando alternativas (F: primero se calcula la secuencia de acciones y luego se ejecutan)

   c. En la etapa de búsqueda se debe decidir qué hacer eligiendo entre las acciones que lleven al estado objetivo (V)

   d. Es necesario percibir para poder determinar el estado en el que se está (V)

6. Los problemas de tipo:

   a. Deterministas y completamente observables son difíciles de resolver (F: son los más simples)

   b. No deterministas y/o parcialmente observables se les suele conocer como problemas de contingencia (V)

   c. Conformados el agente sabe en todo momento donde está (F: el agente puede no saber dónde está)

   d. En los problemas no observables hay que considerar como estado inicial cualquiera del espacio de estados (V)

7. En el algoritmo de búsqueda primero en anchura:

   a. Se expande el nodo de menor profundidad que aún no esté expandido (V)

   b. La frontera incluye los sucesores ordenada por antigüedad (primero los más

UNIVERSITAT DE BARCELONA

Departament de Matemàtica Aplicada i Anàlisi
Gran Via de les Corts Catalanes, 585
08007 Barcelona
Tel . 93 402 16 50, Fax . 93 403 16 01

antiguos) (V)

c. Es óptimo y el espacio no es un problema (F: el espacio es el mayor problema)

d. Es completo sólo cuando el máximo factor de ramificación es finito (V)

8. El algoritmo de de búsqueda:

   a. En grafos es necesario tener una estructura para almacenar los nodos expandidos denominada frontera (F: es la lista de cerrados no la frontera)

   b. De coste uniforme guarda en la frontera nodos pendientes de expandir ordenados por coste decreciente (F: orden creciente)

   c. Primero en profundidad se expande el nodo no expandido más profundo (V)

   d. Primero en profundidad no es completo, pero su complejidad espacial es exponencial (F: es lineal)

9. Los algoritmos de búsqueda informada:

   a. Utilizan información de dominio para dirigir la búsqueda (V)

   b. Utilizan la heurística, una función que para cada nodo, da la distancia a la que se encuentra dicho nodo con respecto al objetivo (F: no la da, la estima)

   c. De tipo greedy son completos pero no óptimos (F: no es ni completa ni óptima)

   d. Usan tanto la distancia del nodo actual al objetivo como el coste desde el estado inicial al actual (F: el greedy o voraz no lo usa)

10. El algoritmo A*

    a. Usa la función de evaluación g(n)=f(n)+h(n) para guiar la búsqueda teniendo en cuenta que g(n) describe el coste estimado de la mejor solución que pase por n (F: es f(n)=g(n)+h(n))

    b. Usa la función h(n) para estimar el coste desde el nodo actual hasta el nodo objetivo n (F: n es el nodo actual, no el objetivo)

    c. Requiere una heurística admisible (es decir, $h(n) \leq h^*(n)$ para todo n) para poder ser óptimo en un árbol (V)

    d. Requiere de una heurística consistente (es decir, f(n) es no decreciente en cualquier camino) para poder ser óptimo en grafos (V).

UNIVERSITAT DE BARCELONA

Departament de Matemàtica Aplicada i Anàlisi
Gran Via de les Corts Catalanes, 585
08007 Barcelona
Tel . 93 402 16 50, Fax . 93 403 16 01

Problema

Tal y como muestra la figura de la izquierda, tenemos tres muñecas rusas geeks, donde la más pequeña es de un Megabyte que no se puede abrir, la mediana es un Gigabyte y la mayor representa un Terabyte. Asumimos que tenemos un robot capaz de meter una muñeca X dentro de otra muñeca Y siempre que: X < Y; X está cerrada y sobre la mesa, e Y está abierta, vacía y sobre la mesa. Se trata pues de que este robot determine la manera de guardar todas las muñecas dentro de la mayor, tal y como se muestra en la parte derecha de la figura. Este robot siempre explora las posibles acciones a hacer con el mismo orden: primero intenta las acciones de abrir muñeca, luego la de meter una muñeca dentro de otra y finalmente la de cerrar muñeca.

Estado inicial

Objetivo

Antes de hacer la programación se solicita:

1. La formalización del problema de búsqueda, definiendo con claridad (de manera formal) el conjunto de estados.
2. Dibujad el árbol que se va abriendo durante la búsqueda en grafos primero en profundidad (o profundidad prioritaria). Representad cada nodo dentro de un cuadrado. Etiquetad las aristas del árbol con la acción que lleva de un estado a otro. Mantened el orden de aristas indicado en el enunciado. Indicad las repeticiones de nodo mediante un cuadrado con contorno de doble línea. Utilizad doble subrayado para resaltar el nombre del estado final.
3. Indicad la solución que retornará

UNIVERSITAT DE BARCELONA

Departament de Matemàtica Aplicada i Anàlisi
Gran Via de les Corts Catalanes, 585
08007 Barcelona
Tel . 93 402 16 50, Fax . 93 403 16 01

1. Estados: representaremos una muñeca $X \in \{M, G, T\}$, $\underline{X}$ denota que la muñeca está cerrada, $\not{X}$ como abierta, $(X_j \supset X_i)$ X_i dentro de X_j, y el estado $s= \{X_i \mid i=1..3, X_i \in \{X, (X_i \supset X_j), \varnothing\}$, representa cómo se encuentran las tres muñecas. Consideraremos X_i, X_j. $X_k \in s$, i,j,k=1..3, $i \neq j \neq k$
   Estado inicial S1= [$\underline{M}$, $\underline{G}$, $\underline{T}$];
   Estado final SF=[$(\underline{T} \supset (\underline{G} \supset \underline{M}))$]
   Acciones posibles: cada acción se aplica sobre una muñeca X_i dentro del estado s.
   - abrir muñeca X_i si $X_i \in \{G, T\}$ y $\underline{X_i}$
   - mover muñeca X_i dentro de X_j, si $X_i < X_j$, $\underline{X_i}$, $\not{X_j}$, no$\exists$ X_k tq $X_k \supset X_j$, no$\exists$ X_k tq $X_k \supset X_i$, no$\exists$ X_k tq $X_j \supset X_k$,, (es decir, X_j no está en ninguna otra muñeca, X_i no está dentro de ninguna otra muñeca y X_j no tiene ninguna otra muñeca dentro)
   - cerrar muñeca X si $X_i \in \{G, T\}$ y $\not{X_i}$

   Función sucesor: S x A -> S, dado un estado, al aplicar una acción posible, se pasa a otro estado: abrir una muñeca cerrada $\underline{X_i} \in s$ pasa a un estado s' en el que la muñeca está abierta $\not{X_i} \in s'$, al meter la muñeca $\underline{X}$ dentro de $\not{Y}$ pasa a $(\not{Y} \supset \underline{X})$, al cerrar una muñeca abierta $\not{X_i}$ pasa a cerrada $\underline{X_i}$.
   Es decir si consideramos estos estados:
   $s_C = \{X_i \mid \exists\ X_i = \underline{X}\}$
   $s_A = \{X_i \mid \exists\ X_i = \not{X}\}$
   $s_{M1} = \{X_i \mid \exists\ X_i = \underline{X}, \exists\ X_j = \not{X}\ , X_i < X_j$, no$\exists$ X_k tq $X_k \supset X_j$, no$\exists$ X_k tq $X_k \supset X_i$, no$\exists$ X_k tq $X_j \supset X_k\}$
   $s_{M2} = \{X_i \mid \exists\ X_k = (\underline{X_i} \subset \not{X_j})\}$
   entonces tenemos que
   $succ(s_C, \text{abr } X_i) = s_A$
   $succ(s_{M1}, \text{mov } (X_i , X_j)) = s_{M2}$
   $succ(s_A, \text{cer } X_i) = s_C$

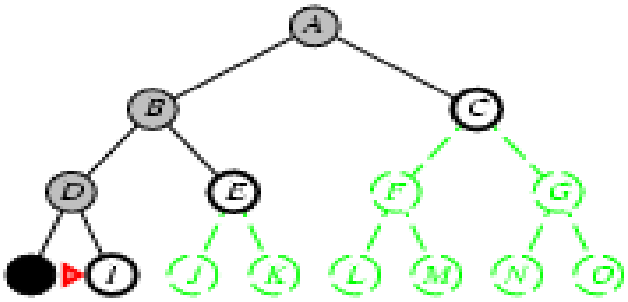
NOTA: Este pseudocódigo y la figura siguiente no los pedía, los añado a modo de aclaración

```
función BÚSQUEDA-GRAFOS(problema,frontera) devuelve una solución o fallo
    cerrados ← un conjunto vacío
    frontera ← INSERTA(HACER-NODO(ESTADO-INICIAL[problema]), frontera)
    bucle hacer
        si VACIA? (frontera) entonces devolver fallo
        nodo ←SACAR-BORRANDO-PRIMERO(frontera)
        si TEST-OBJETIVO[problema] aplicado al ESTADO[nodo] es cierto
            entonces devolver SOLUCION(nodo)
        si ESTADO[nodo] no está en cerrados entonces
            añadir ESTADO[nodo] a cerrados
            frontera ← INSERTAR-TODO(EXPANDIR(nodo, problema), frontera)
```

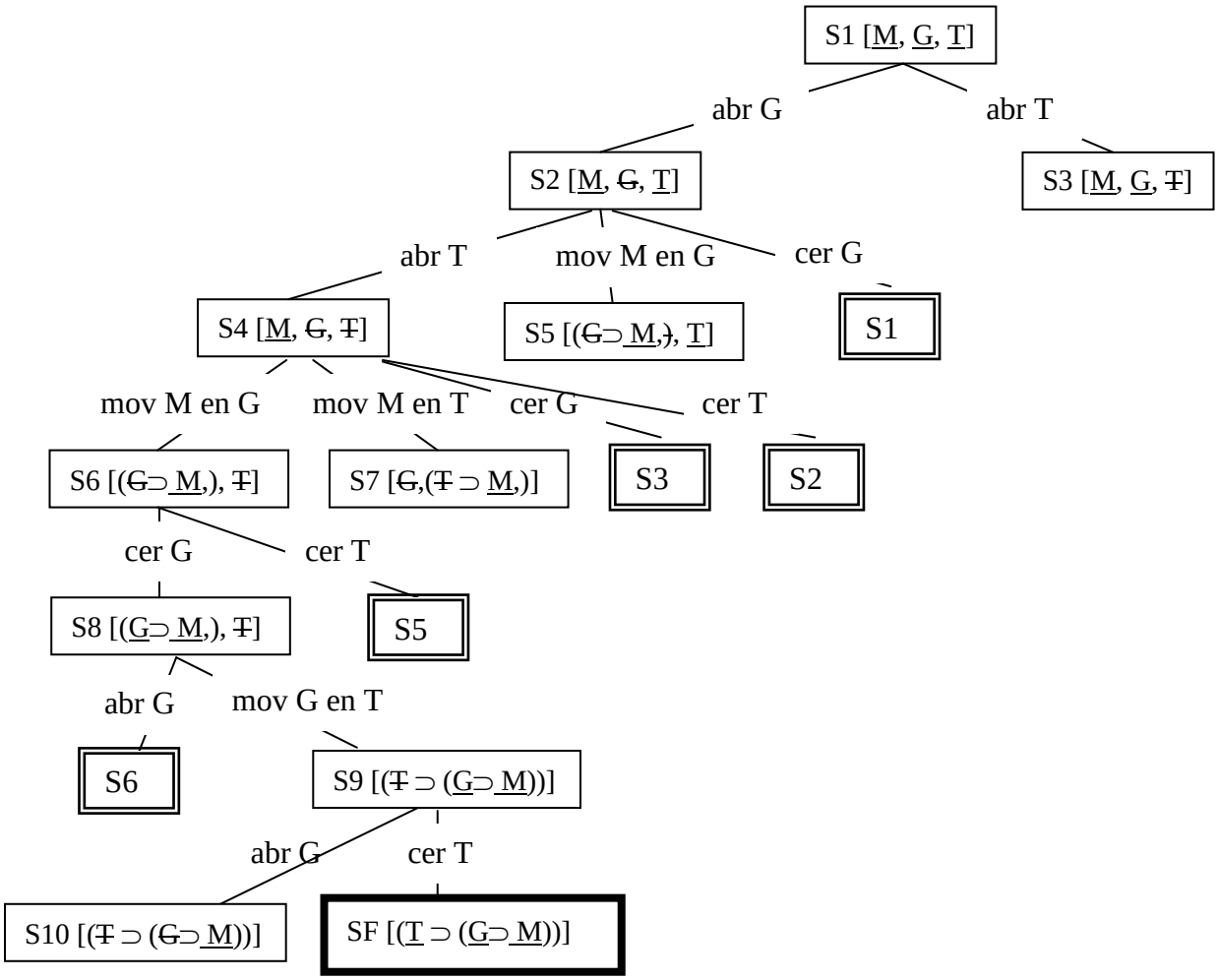
UNIVERSITAT DE BARCELONA

Departament de Matemàtica Aplicada i Anàlisi
Gran Via de les Corts Catalanes, 585
08007 Barcelona
Tel . 93 402 16 50, Fax . 93 403 16 01

Ejemplo DFS:



2. Árbol DFS: orden acciones: abr (abrir), mov (mover), cer (cerrar), orden muñecas: M, G, T



3. solución: secuencia de acciones: abrir G, abrir T, mover M en G, cerrar G, mover G en T, cerrar T.