Inteligencia Artificial
Hoja 1: Búsqueda y juegos

Publicación: 2018/02/21
Entrega: 2018/02/26, 11:00
Resolución en clase: 2018/02/26

---

1. En el siguiente problema vamos a resolver un problema del puzle *Master Mind*, en una versión simplificada. El puzle consiste en adivinar una secuencia de 3 colores escogidos de entre 4 posibles: Verde (V), Magenta (M), Rojo (R), Azul (A)
Las reglas del juego son las siguientes:

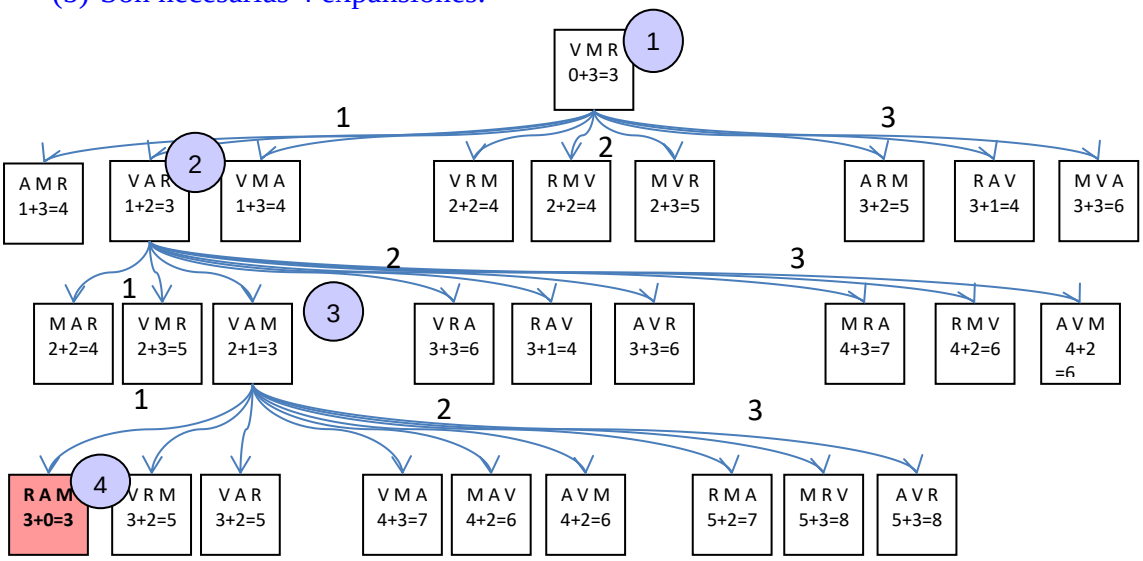
- El estado inicial es la secuencia VMR.
- No se pueden repetir colores en la secuencia.
- La secuencia oculta que buscamos es RAM.
- Los operadores aplicables a cada estado son (**se deben aplicar en este orden, comenzando de izquierda a derecha en la secuencia y respetando el orden de colores V, M, R, A**):
  (1) Cambiar un color, dejando fijos los otros dos: coste 1
  Ejemplo: VMR → AMR
  (2) Dejar fijo un color, permutando los otros dos: coste 2
  Ejemplo: VMR → VRM
  (3) Cambiar un color y permutar los otros dos: coste 3.
  Ejemplo: VMR → ARM

a) Propón una heurística monótona para este problema.
b) Desarrolla el árbol de búsqueda empleando el algoritmo A* sin eliminación de estados repetidos. Para cada nodo, indicar el orden de expansión y los valores de coste $g+h=f$
c) ¿Es A* sin eliminación de estados repetidos con la heurística propuesta óptima?
d) ¿Sería A* con eliminación de estados repetidos con la heurística propuesta óptima?

SOLUCIÓN

(a) Contar el número de piezas que tienen color no incluido en la meta más número de piezas descolocadas. La h resulta de la solución de un problema relajado, en el que reemplazamos directamente las piezas incorrectas por otras correctas. En consecuencia es monótona.

(b) Son necesarias 4 expansiones.



(c) Dado que la heurística es monótona, también es admisible.
A* con heurística admisible, sin eliminar estados repetidos = óptima

(d) A* con heurística monótona, eliminando estados repetidos = óptima.

2. Consideremos una variante del nim, en el que, partiendo de una pila de 7 monedas, dos jugadores toman turnos para realizar movimiento. En cada turno el jugador elige una pila de entre las existentes y la divide en **dos** subpilas **no vacías** y **de tamaños diferentes**. Pierde el jugador que se queda sin movimientos.

Dibuja el árbol de juego completo

- Las jugadas que generan subpilas de tamaños más dispares se generan primero.
- En el despliegue del árbol es necesario que tengas en cuenta las simetrías para evitar que este sea excesivamente grande.
- Etiqueta los terminales con el valor de la función de utilidad respecto a MAX.

De acuerdo con la formalización realizada

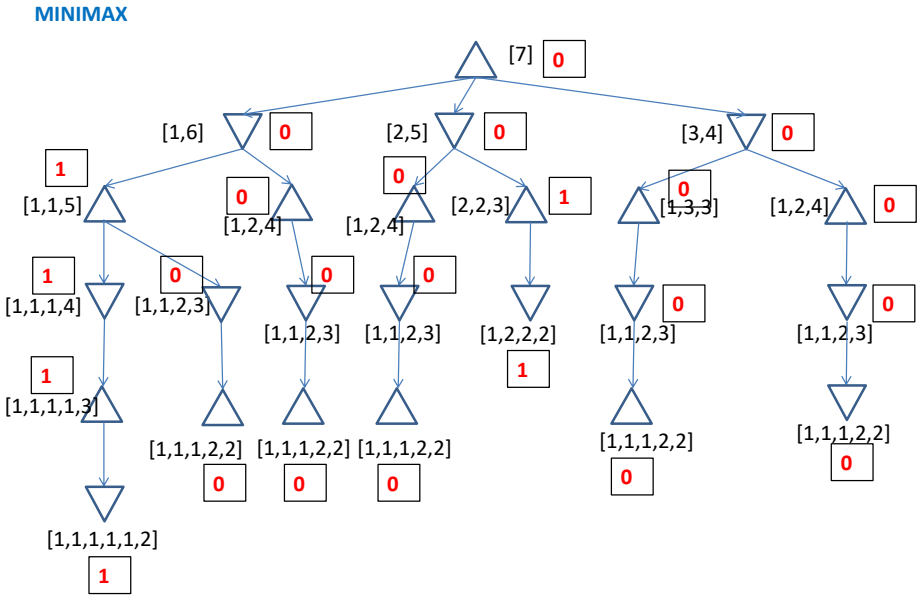
a. ¿Es un juego de suma cero, de suma constante, o de suma no nula y no constante?
b. En caso de que el oponente sea óptimo ¿Cuál sería la estrategia óptima para el jugador que inicia el juego?
c. Sobre el árbol de juego desplegado aplica el algoritmo minimax.
   De acuerdo con este algoritmo, ¿cuál sería el valor minimax en la raíz del árbol? ¿cuál sería la jugada óptima?
d. Sobre una copia del árbol de juego, aplica algoritmo minimax con poda alpha-beta. En esta copia incluye únicamente los nodos visitados.
   De acuerdo con este algoritmo, ¿cuál sería el valor minimax en la raíz del árbol? ¿cuál sería la jugada óptima?

SOLUCIÓN:

a. ¿En un juego de suma cero, de suma constante, o de suma no nula y no constante?
Con la formalización realizada, es un juego de suma constante.

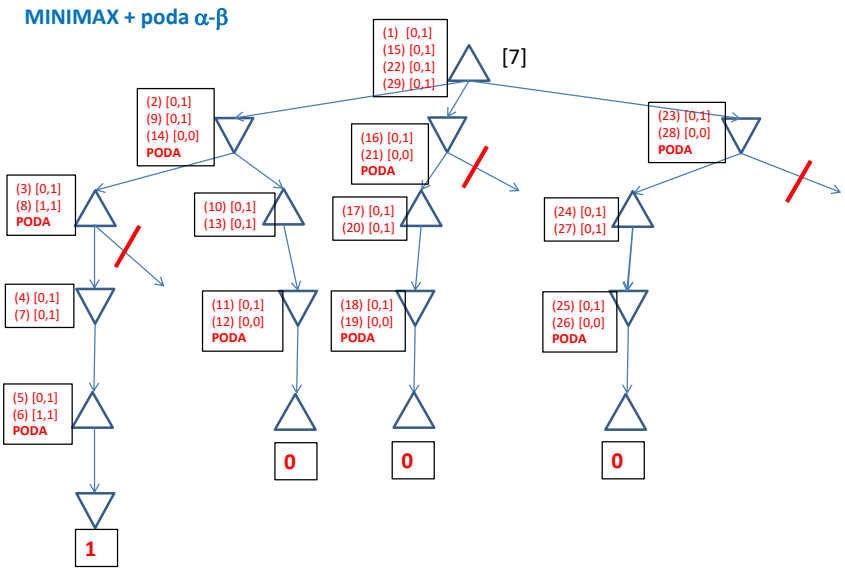
b. En caso de que el oponente sea óptimo ¿Cuál sería la estrategia óptima para el jugador que inicia el juego?
La estrategia óptima en el caso de que el oponente sea óptimo es la resultante de aplicar el algoritmo minimax.

c. Sobre el árbol de juego desplegado aplica el algoritmo minimax.
De acuerdo con este algoritmo, ¿cuál sería el valor minimax en la raíz del árbol?
¿Cuál sería la jugada óptima?



Valor minimax en la raíz: 0
Jugada óptima: Cualquiera de las tres

d. Sobre una copia del árbol de juego, aplica algoritmo minimax con poda alpha-beta. En esta copia incluye únicamente los nodos visitados.
De acuerdo con este algoritmo, ¿cuál sería el valor minimax en la raíz del árbol?
¿Cuál sería la jugada óptima?



Valor minimax en la raíz: 0
Jugada óptima: Cualquiera de las tres

3. En un bosque de pinos que estamos gestionando se ha detectado una plaga de orugas. Es necesario enviar un equipo para fumigar y limpiar el bosque en el mínimo tiempo posible.
El estado del bosque es el siguiente:

0	1
2	X

X: Zona rocosa (inaccesible por tierra y libre de la plaga)
0: Zona sin plaga
1: Zona con población de oruga baja
2: Zona con población de oruga alta

N
O E
S

El equipo de fumigación posee un mapa en el que se indican únicamente los sectores del bosque. El grado de infección solo se puede conocer sobre el terreno.

- Para eliminar la plaga en una zona con población de oruga baja, necesitamos realizar una única operación de limpieza. Para eliminar la plaga en una zona con población de oruga alta, necesitamos realizar dos operaciones de limpieza con una separación de, al menos, 4 horas.
- Para comenzar su tarea, el equipo puede acceder al bosque en cualquier sector, ya que será transportado en helicóptero. Para generar el árbol de búsqueda el orden de exploración para el acceso es el de las agujas del reloj, comenzando con el sector al noroeste. Es decir: NO, NE, SE, SO.
- Para realizar la limpieza, tienen que hacerlo a pie. El tiempo para realizar una operación de limpieza de un sector es 1 hora.
- El tiempo para desplazarse entre sectores adyacentes es de 2 horas en horizontal (oeste ↔ este) o vertical (norte ↔ sur) y 3 horas en diagonal (sudoeste ↔ nordeste, sudeste ↔ nororeste). Para generar el árbol de búsqueda el orden de exploración de desplazamientos es el de las agujas del reloj, empezando por el movimiento que produce un desplazamiento hacia el norte. Es decir, el orden de exploración es N, NE, E, SE, S, SO, O, NO.
- Una vez que el equipo ha eliminado la plaga en el bosque el equipo será transportado de nuevo por helicóptero desde el sector en el que se encuentre.

a. Formaliza los estados de búsqueda.

Utilizando la formalización propuesta:

b. Especifica las acciones para generar sucesores, las precondiciones necesarias para aplicarlas y su coste.
c. Especifica el estado inicial.
d. Especifica el test objetivo.
e. Define una heurística admisible no trivial para resolver el problema y demuestra su admisibilidad.
f. Detalla el árbol generado por A* con eliminación de estados repetidos. Indica para cada nodo los valores g + h = f y el orden en el que el **intento de exploración** se realiza (es decir, los nodos repetidos y la meta también reciben numeración). **En caso de que haya empates, se elegirá primero en la exploración el nodo que haya sido generado antes.**
g. De acuerdo con el resultado obtenido, ¿cuál es la estrategia óptima? ¿cuál es el coste óptimo?
h. ¿Garantiza A* con heurística admisible y eliminación de estados repetidos encontrar la solución óptima?
i. ¿Encuentra A* la solución óptima en este ejemplo? Justifica tu respuesta.

SOLUCIÓN:

a. Formaliza los estados de búsqueda.
Cada estado se especifica indicando la posición del equipo de limpieza y el grado de infección en cada uno los sectores

A_{11}	A_{12}
A_{21}	X

$0 \leq A_{11} \leq 1;\ 0 \leq A_{12} \leq 1;\ 0 \leq A_{21} \leq 2;$

b. Especifica las acciones para generar sucesores y su coste.

Se pueden realizar cinco tipos de acciones:

- Comenzar la exploración [coste 0]: El
  - Precondiciones: Solo se puede aplicar en el estado inicial.

- Realizar un desplazamiento a alguna de las celdas adyacentes que sean accesibles.
  - Precondiciones: Sector en el que se encuentra el equipo de limpieza libre de plagas (nivel 0) + algún de los otros sector por limpiar
  - Movimientos N, S, E, O [Coste 2]
  - Movimientos NE,NO SE, SO [Coste 3]

- Limpiar el sector en la que se encuentra el equipo de limpieza sin desplazar [Coste 1].
  - Precondiciones: Sector en el que se encuentra el equipo de limpieza con una población baja de orugas (nivel 1) + restos de sectores libres de plaga (nivel 0)

- Realizar un desplazamiento a alguna de las celdas adyacentes que sean accesibles.
  - Precondiciones: Sector en el que se encuentra el equipo de limpieza libre de plagas (nivel 0) + algún de los otros sectores por limpiar
  - Movimientos N, S, E, O [Coste 2]
  - Movimientos NE,NO SE, SO [Coste 3]

- Limpiara la celda en la que se encuentra el equipo y realizar un desplazamiento a alguna de las celdas adyacentes que sean accesibles.
  - Precondiciones: Sector en el que se encuentra el equipo de limpieza con población de orugas (nivel > 0) + algún de los otros sectores por limpiar
  - Limpieza + movimientos N, S, E, O [Coste 1 + 2 = 3]
  - Limpieza + movimientos NE,NO SE, SO [Coste 1 + 3= 4]

c. Especifica el estado inicial.

0	1
2	X

El equipo de limpieza aún no ha accedido al bosque.

d. Especifica el test objetivo.

0	0
0	X

El equipo de limpieza puede estar en cualquier sector.

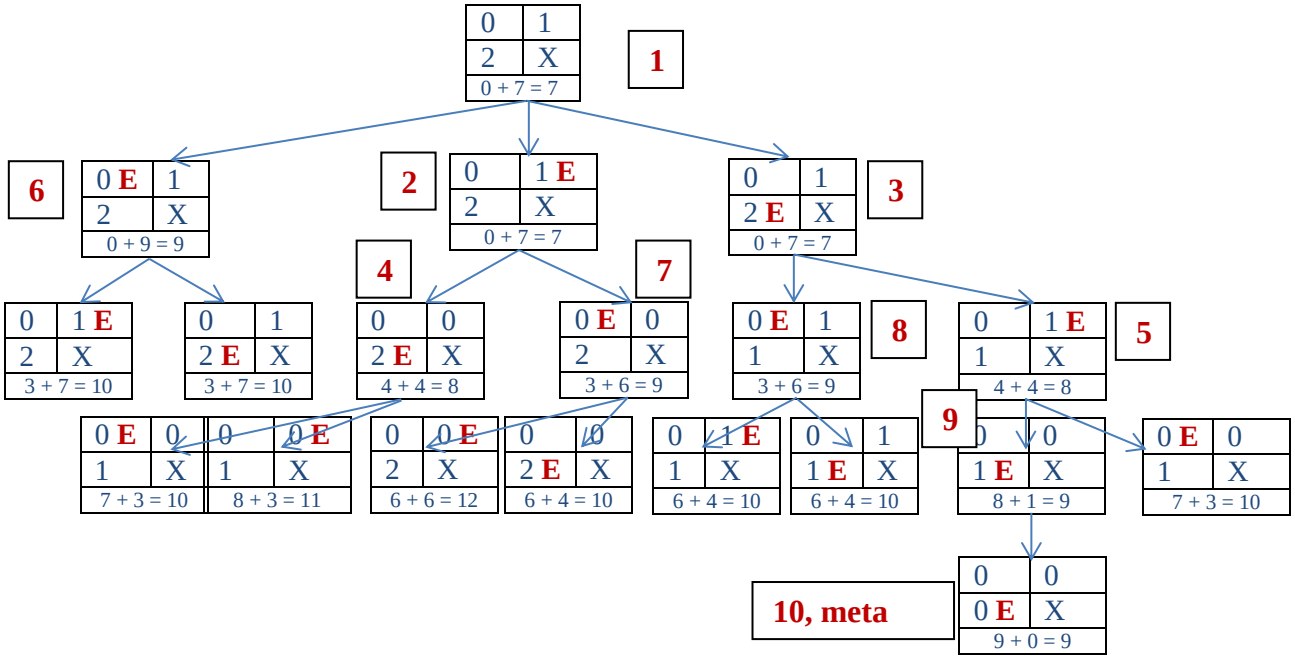
e. Define una heurística admisible no trivial para resolver el problema y demuestra su admisibilidad.

Por relajación, suponer que:

- cada sector (i,j) requiere A_{ij} limpiezas [coste de operación de limpieza 1],
- el número de desplazamientos es igual a $(A_{11}+A_{12}+A_{21})$ en caso de que el sector en el que se encuentra el equipo está libre de orugas, $(A_{11}+A_{12}+A_{21}) - 1$, si es el estado inicial o el sector en el que se encuentra el equipo de limpieza está infestado.
- en el problema relajado, el coste de un desplazamiento es igual al coste mínimo de los desplazamiento posibles en el problema original [coste de desplazamiento 2]

(1+2)*(A_{11}+A_{12}+A_{21}) [– 2, si es el estado inicial o si el sector en el que se encuentra el equipo de limpieza está infestado, ya que no es necesario desplazarse a ese sector para realizar la operación de limpieza]

f. Detalla el árbol generado por A* con eliminación de estados repetidos. Indica para cada nodo los valores g + h = f y el orden en el que el **intento de exploración** se realiza (es decir, los nodos repetidos y la meta también reciben numeración). **En caso de que haya empates, se elegirá primero en la exploración el nodo que haya sido generado antes.**



g. De acuerdo con el resultado obtenido, ¿cuál es la estrategia óptima? ¿cuál es el coste óptimo?

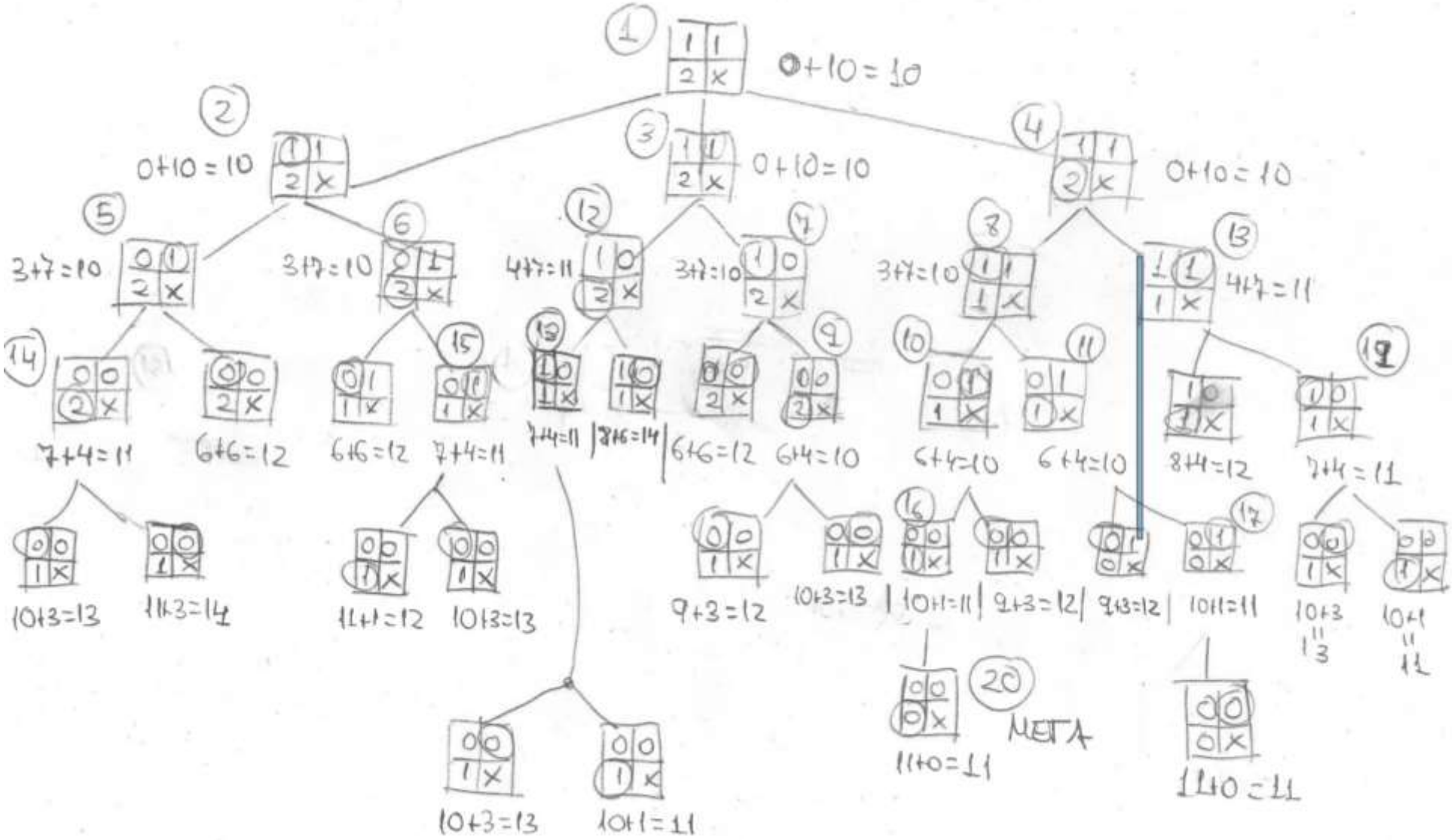
SO [limpia, 1] →, 3 NE [limpia, 1] →, 3 SO [limpia, 1]. Coste total mínimo: 9

h. ¿Garantiza A* con heurística admisible y eliminación de estados repetidos encontrar la solución óptima?

No, A* + eliminación de estados repetidos + h admisible no es necesariamente óptima.

i. ¿Encuentra A* la solución óptima en este ejemplo? Justifica tu respuesta.

Sí, porque la heurística es monótona (es la solución óptima de un problema relajado):
A* + eliminación de estados repetidos + h monótona = óptima.



SO [limpia,1] →,2 NO [limpia,1] →,2 NE [limpia,1] →,3 SO [limpia,1]. Coste total mínimo: 11

4. Se parte de la siguiente situación en el juego "tres en raya" en la que le toca jugar al jugador con ficha X:

O		
X	O	
	O	X

El jugador utiliza el algoritmo minimax con poda alfa-beta hasta profundidad 3.
Se debe utilizar la función de evaluación:

1. Estado terminal ganador para el jugador que tiene el turno: $+\infty$
2. Estado terminal ganador para el jugador contrario: $-\infty$
3. Estado terminal en tablas: 0
4. En cualquier otro caso, la diferencia entre el nº de posibles líneas (filas, columnas, diagonales) que podría completar el jugador que tiene el turno menos el nº de líneas que podría completar su adversario, suponiendo que ambos pueden colocar en el tablero tantas fichas como deseen.

- Dibujad el árbol que se genera indicando los valores obtenidos en los nodos hoja por la función de evaluación. **Se debe dibujar solo la parte del árbol que se genera** (por lo tanto, no es necesario dibujar los nodos del árbol de juego a los que no se accede debido a poda)..
- Se debe especificar cómo evolucionan en cada nodo los **valores de alfa y beta**, **numerando** las distintas actualizaciones del intervalo, e indicar **dónde en el árbol se realiza poda**.

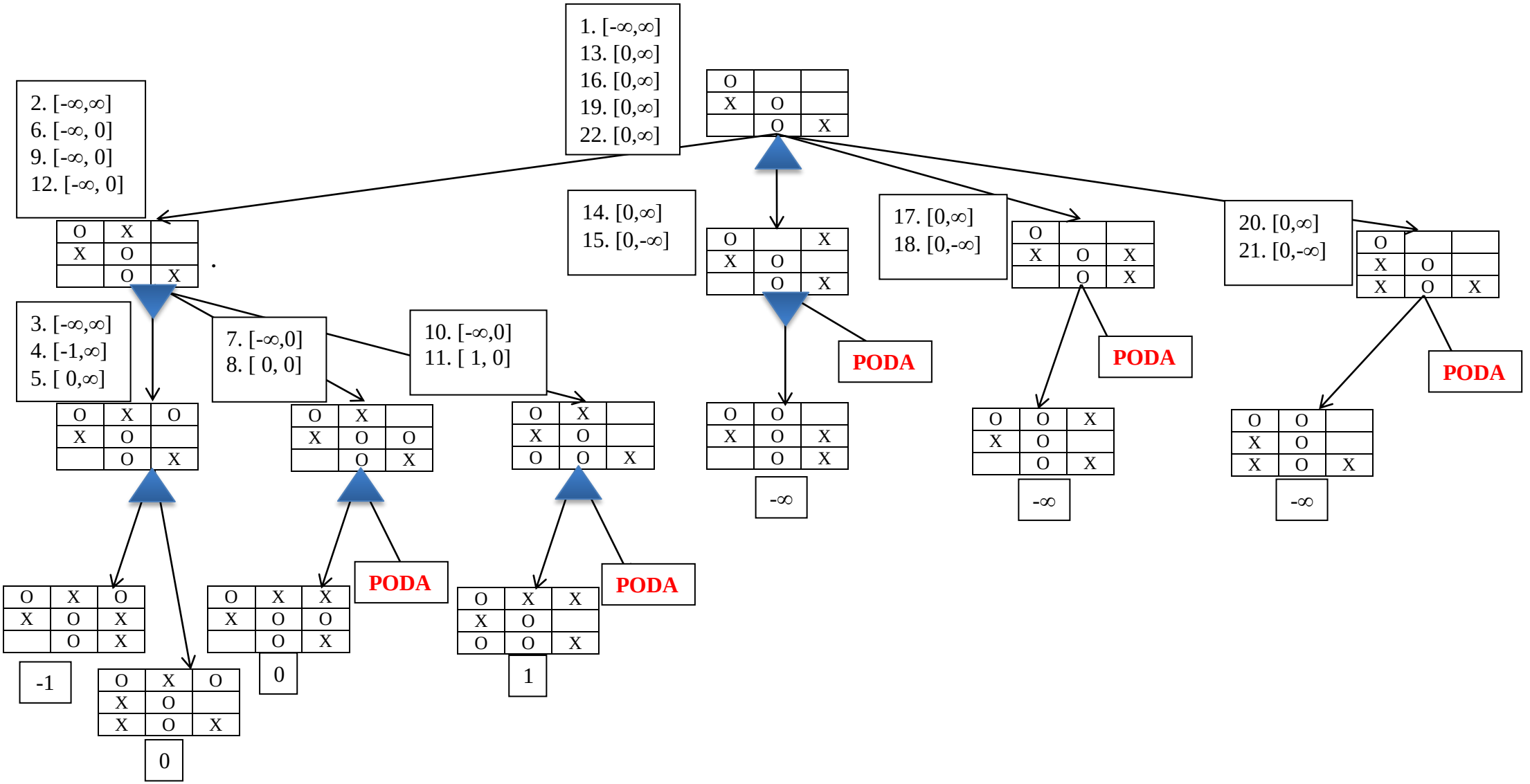
ATENCIÓN: Para la generación de los sucesores de un nodo dado, se debe considerar el siguiente orden: **primero** las **filas de arriba hacia abajo**, y después, dentro de la misma fila, las **columnas de izquierda a derecha**. Por ejemplo, el primer nodo sucesor del nodo de partida sería:

O	X	
X	O	
	O	X

Con todas las indicaciones dadas:

a) ¿Cuál es el valor minimax para el jugador que tiene el turno? ¿Qué movimiento debería a realizar dicho jugador? Razona tu respuesta.

b) ¿Ha influido de alguna manera el orden indicado en la elección del movimiento del jugador? ¿y en la eficiencia de la poda? Razona tu respuesta.



a) El valor minimax es 0. El jugador 'x' (MAX) debería hacer la jugada de la izquierda.

b) El oden de exploración podría influir en el caso de que hubiera varias jugadas óptimas. En este caso, solo hay una jugada óptima, por lo que el orden de exploración no influye en la elección.
No obstante, si ha influido en la eficiencia de la poda: El hecho de haber explorado primero la mejor jugada posible para MAX y para MIN, permite hacer más poda.

5. El objetivo es equilibrar balanza de dos brazos, sobre uno de cuyos platos se coloca un objeto de 3 Kg. Para equilibrar la balanza se pueden utilizar pesas de 4Kg y de 5 Kg.
   Las acciones consisten en colocar pesas en alguno de los platos de la balanza. El coste de una acción es el peso añadido.

De Toby Hudson –
Trabajo propio, CC BY-SA 3.0,
https://commons.wikimedia.org/w/index.php?curid=277

   a. Formaliza el problema, **indicando claramente**:
      i. **Codificación de los estados de búsqueda**
      Define una estructura que permita identificar de manera unívoca cada estado de búsqueda.
      ii. **Estado inicial.**
      Especifica el valor inicial de los campos de la estructura definida.
      iii. **Test** para determinar si se ha alcanzado el **objetivo**.
      Define una función booleana que tome como argumento la estructura que caracteriza un estado de búsqueda y permita determinar si este estado corresponde al objetivo.
      iv. **Codificación de las acciones.** Utiliza para ello la codificación Propuesta para los estados de búsqueda.
      <nombre_acción>: <estado_inicial> → <estado_final>

   b. Define una **heurística monótona** para el problema. Justifica la monotonicidad de la heurística.
   c. ¿**Es la heurística definida admisible**? Justifica la respuesta.
   d. Encuentra una **solución del puzle mediante A* con la heurística propuesta y eliminación de estados repetidos**. Para ello desplegad el árbol de búsqueda, etiquetando cada nodo n con los valores g(n) + h(n) = f(n). Indicad asimismo el orden en el que se consideran los nodos para ser o bien expandidos o bien ser eliminados por corresponder a estados repetidos. En caso de empate, se expandirá primero el nodo más antiguo.

   Puede que sea más cómodo dibujar el árbol en una **hoja con formato apaisado**, con el fin de tener suficiente espacio para el etiquetado.
   e. **Indica la solución encontrada y su coste**.
   f. ¿**Es la solución encontrada óptima**? Justifica la respuesta.

SOLUCIÓN:

a. Formalizad el problema, **indicando claramente**:

i. Codificación de los estados de búsqueda

Definid una estructura que permita identificar de manera unívoca cada estado de búsqueda.

$[o_3, n_{4L}, n_{5L} \mid n_{4R}, n_{5R}]$
o_3 = Objeto de 3 Kg (suponemos que está en el plato izquierdo)
n_{4L} = Número de pesas de 4 Kg en el plato izquierdo
n_{4R} = Número de pesas de 4 Kg en el plato derecho
n_{5L} = Número de pesas de 5 Kg en el plato izquierdo
n_{5R} = Número de pesas de 5 Kg en el plato derecho

Restricciones:

$[n_{4L} \neq 0] \Rightarrow [n_{4R} = 0]$, $[n_{4R} \neq 0] \Rightarrow [n_{4L} = 0]$
$[n_{5L} \neq 0] \Rightarrow [n_{5R} = 0]$, $[n_{5R} \neq 0] \Rightarrow [n_{5L} = 0]$

ii. Estado inicial.

Especificad el valor inicial de los campos de la estructura definida.
$[o_3, 0_4, 0_5 \mid 0_4, 0_5]$

iii. **Test** para determinar si se ha alcanzado el **objetivo**.

Definid una función booleana que tome como argumento la estructura que caracteriza un estado de búsqueda y permita determinar si este estado corresponde al objetivo.

Test($[o_3, n_{4L}, n_{5L} \mid n_{4R}, n_{5R}]$) := $[3 + 4\, n_{4L} + 5\, n_{5L} == 4\, n_{4R} + 5 n_{5R}]$

iv. **Codificación de las acciones.** Utilizad para ello la codificación de los estados.
<nombre_acción>: <estado_inicial> → <estado_final>

Añade pesa de 4Kg en lado izdo.	**A_{4L}:**	**$[o_3, n_{4L}, 0 \mid 0, n_{5R}] \rightarrow [o_3, n_{4L} + 1, 0 \mid 0, n_{5R}]$**
Añade pesa de 4Kg en lado dcho.	**A_{4R}:**	**$[o_3, 0, n_{5L} \mid n_{4R}, 0] \rightarrow [o_3, 0, n_{5L} \mid n_{4R} + 1, 0]$**
Añade pesa de 5Kg en lado izdo.	**A_{5L}:**	**$[o_3, 0, n_{5L} \mid n_{4R}, 0] \rightarrow [o_3, 0, n_{5L} + 1 \mid n_{4R}, 0]$**
Añade pesa de 5Kg en lado dcho.	**A_{5R}:**	**$[o_3, n_{4L}, 0 \mid 0, n_{5R}] \rightarrow [o_3, n_{4L}, 0 \mid 0, n_{5R} + 1]$**

g. Define una heurística monótona para el problema. Justifica la monotonicidad de la heurística.
$h([o_3, n_{4L}, n_{5L} \mid n_{4R}, n_{5R}]) := [3 + 4\, n_{4L} + 5\, n_{5L}] - [4\, n_{4R} + 5\, n_{5R}]$

La heurística es monótona ya que es la solución de un problema relajado.

h. ¿Es la heurística definida admisible? Justifica la respuesta.
La heurística es admisible ya que toda heurística monótona es admisible.

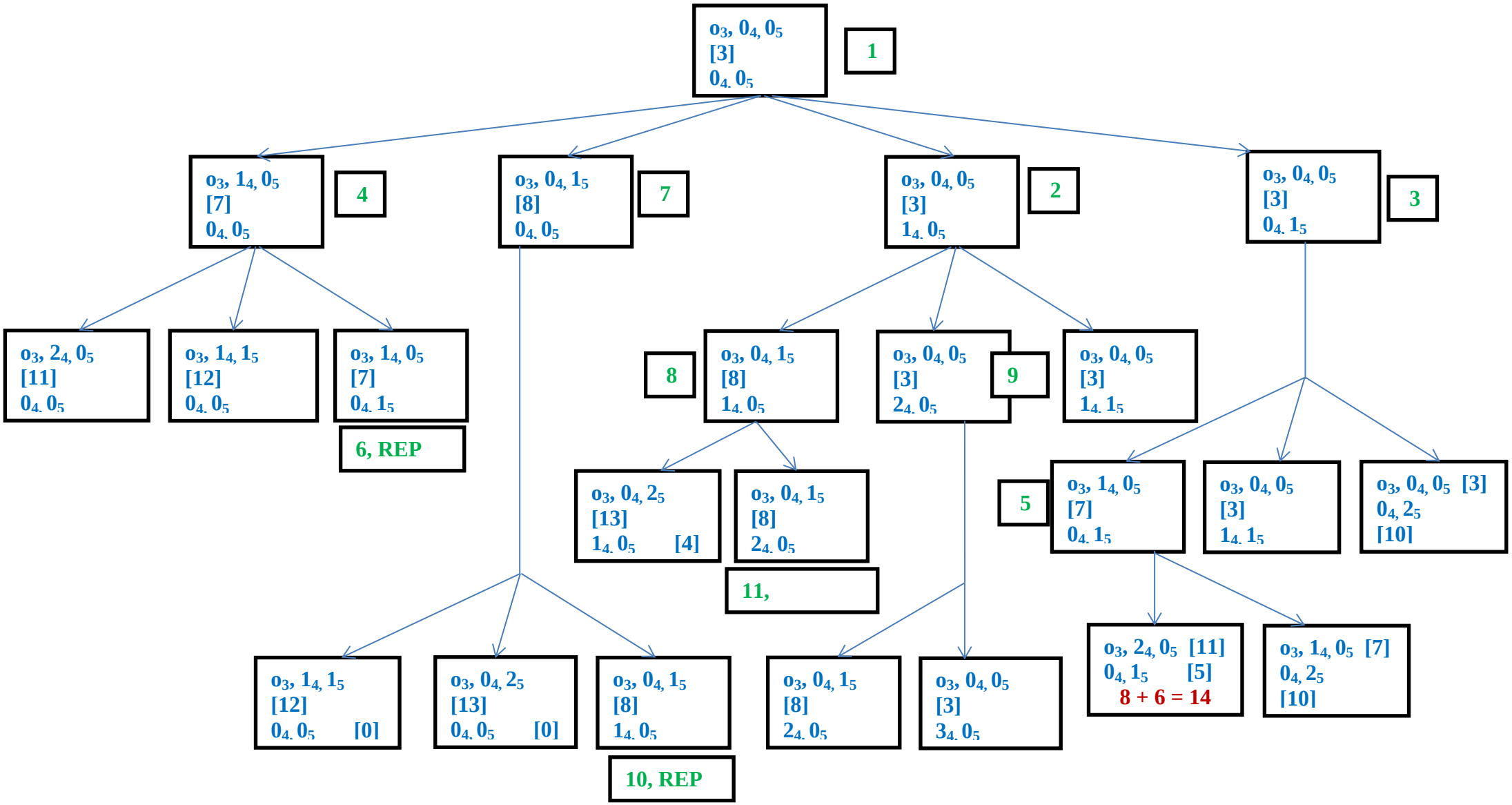
i. Encuentra una solución del puzle mediante A* con la heurística propuesta y eliminación de estados repetidos. Para ello desplegad el árbol de búsqueda, etiquetando cada nodo n con los valores g(n) + h(n) = f(n). Indicad asimismo el orden de expansión de los nodos y qué nodos han sido eliminados por corresponder a estados repetidos. En caso de empate, se expandirá primero el nodo más antiguo.
Ver árbol de búsqueda en la siguiente página.

j. Indica la solución encontrada y su coste.

$[o_3, 0_4, 0_5 \mid 0_4, 0_5] \rightarrow [o_3, 0_4, 1_5 \mid 0_4, 0_5] \rightarrow [o_3, 0_4, 1_5 \mid 1_4, 0_5] \rightarrow [o_3, 0_4, 1_5 \mid 2_4, 0_5]$
Coste = 1×5 + 2×4 = 13

k. ¿Es la solución encontrada óptima? Justifica la respuesta.

Sí. A* con eliminación de estados repetidos + heurística monótona = óptima.


$o_3, 0_4, 0_5$ [3] $0_4, 0_5$
1
$o_3, 1_4, 0_5$ [7] $0_4, 0_5$
4
$o_3, 0_4, 1_5$ [8] $0_4, 0_5$
7
$o_3, 0_4, 0_5$ [3] $1_4, 0_5$
2
$o_3, 0_4, 0_5$ [3] $0_4, 1_5$
3
$o_3, 2_4, 0_5$ [11] $0_4, 0_5$
$o_3, 1_4, 1_5$ [12] $0_4, 0_5$
$o_3, 1_4, 0_5$ [7] $0_4, 1_5$
6, REP
8
$o_3, 0_4, 1_5$ [8] $1_4, 0_5$
$o_3, 0_4, 0_5$ [3] $2_4, 0_5$
9
$o_3, 0_4, 0_5$ [3] $1_4, 1_5$
$o_3, 0_4, 2_5$ [13] $1_4, 0_5$ [4]
$o_3, 0_4, 1_5$ [8] $2_4, 0_5$
11,
5
$o_3, 1_4, 0_5$ [7] $0_4, 1_5$
$o_3, 0_4, 0_5$ [3] $1_4, 1_5$
$o_3, 0_4, 0_5$ [3] $0_4, 2_5$ [10]
$o_3, 1_4, 1_5$ [12] $0_4, 0_5$ [0]
$o_3, 0_4, 2_5$ [13] $0_4, 0_5$ [0]
$o_3, 0_4, 1_5$ [8] $1_4, 0_5$
10, REP
$o_3, 0_4, 1_5$ [8] $2_4, 0_5$
$o_3, 0_4, 0_5$ [3] $3_4, 0_5$
$o_3, 2_4, 0_5$ [11] $0_4, 1_5$ [5] 8 + 6 = 14
$o_3, 1_4, 0_5$ [7] $0_4, 2_5$ [10]

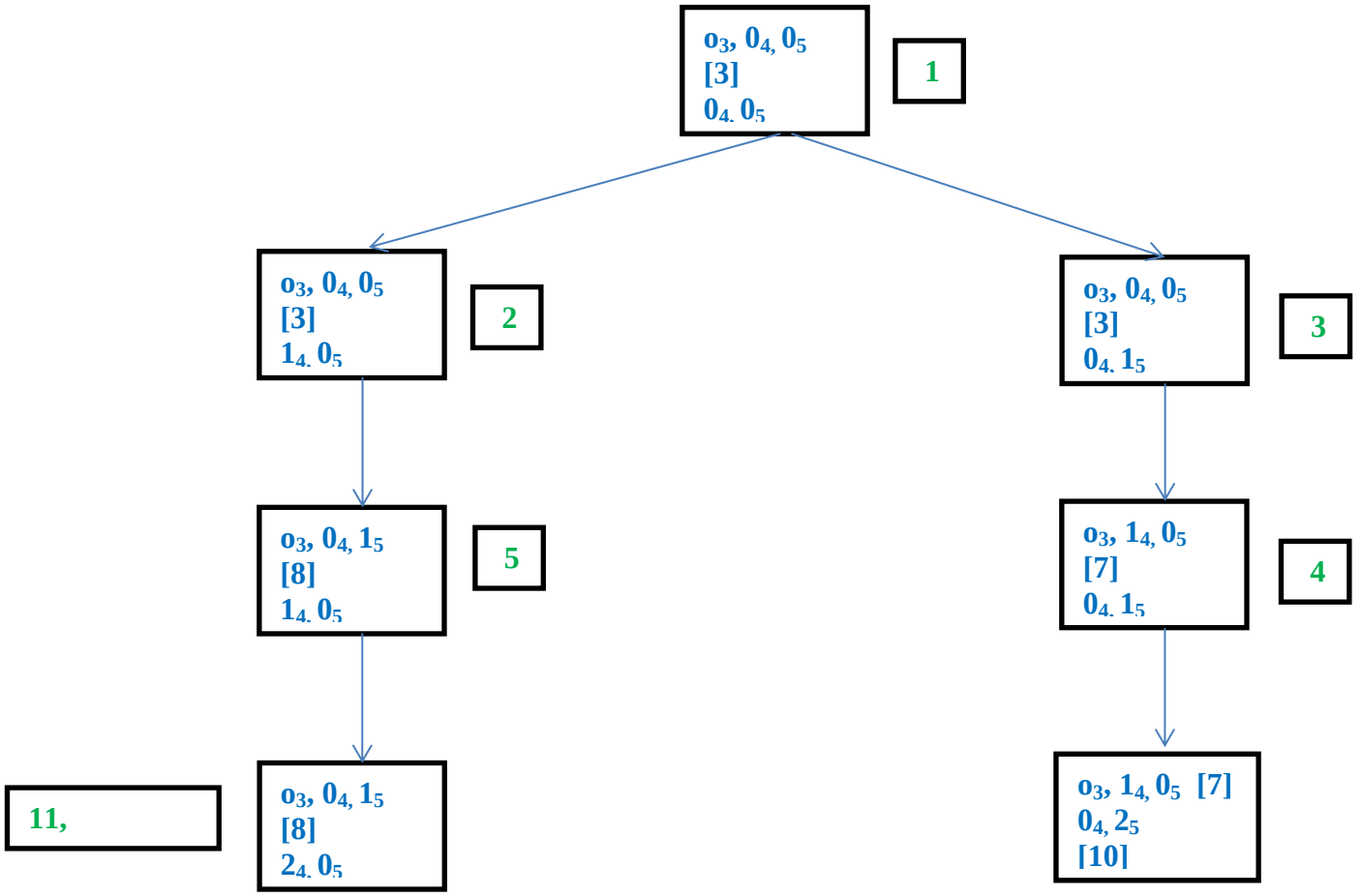
Analizando con más detalle el problema, uno observa que la forma de la solución tiene que ser o bien de la forma

$[o_3, n_4, 0_5 \mid 0_4, n_5]$
$[o_3, 0_4, n_5 \mid n_4, 0_5]$

Las acciones posibles a partir de un estado serían

$[o_3, n_4, 0_5 \mid 0_4, n_5] \rightarrow [o_3, n_4 + 1, 0_5 \mid 0_4, n_5]$ Si $[3 + 4\,n_4] < [5\,n_5]$
$[o_3, n_4, 0_5 \mid 0_4, n_5] \rightarrow [o_3, n_4, 0_5 \mid 0_4, n_5 + 1]$ Si $[3 + 4\,n_4] > [5\,n_5]$
$[o_3, 0_4, n_5 \mid n_4, 0_5] \rightarrow [o_3, 0_4, n_5 + 1 \mid n_4, 0_5]$ Si $[3 + 5\,n_5] < [4\,n_4]$
$[o_3, 0_4, n_5 \mid n_4, 0_5] \rightarrow [o_3, 0_4, n_5 \mid n_4 + 1, 0_5]$ Si $[3 + 5\,n_5] > [4\,n_4]$

Finalmente, el árbol de búsqueda sería



6. Consideremos el siguiente juego en el que dos jugadores, A y B que alternan turnos para colocar fichas numeradas sobre la mesa. La fichas que inicialmente poseen los jugadores son:

   Jugador A: 2, 4, 5
   Jugador B: 2, 3, 6

   Las fichas son visibles a ambos jugadores.
   Comienza colocando una ficha sobre la mesa el jugador A. A partir de ese momento, B y A alternan sus turnos. En cada turno, el jugador correspondiente debe colocar sobre la mesa una ficha con un valor igual o superior a las que están sobre la mesa. El juego termina cuando ninguno de los dos jugadores puede colocar más fichas. La puntuación final de la partida es la diferencia entre la suma de los valores de las fichas que tienen los jugadores en su mano. Pierde aquel jugador que se quede con fichas de mayor valor numérico en la mano.

   IMPORTANTE: Proporciona respuesta para cada uno de estos apartados

   (i) ¿Qué tipo de juego es?
      a. Determinista / aleatorio
      b. Información completa / información no completa
      c. Suma cero / suma constante / suma variable

   (ii) Para este tipo de juego, suponiendo que tanto A como B realizan elecciones óptimas ¿cuál es el algoritmo que determina cuál es la jugada óptima?

   (iii) Desplegad de manera completa el árbol de juego, indicando en cada nodo el estado del juego. Etiquetad los nodos terminales con el valor de la función de utilidad.
   IMPORTANTE: Puede que sea más cómodo dibujar el árbol en una hoja con formato apaisado, con el fin de tener suficiente espacio para el etiquetado.

   (iv) Sobre el árbol desplegado, aplicad el algoritmo minimax, indicando claramente el valor minimax en cada uno de los nodos.
      a. ¿Cuál es el valor minimax en la raíz del árbol de juego?
      b. ¿Cuál es la jugada óptima para A?
      c. ¿Cuál es la secuencia de jugadas óptimas?
      d. ¿Qué jugador gana la partida?

   (v) Aplicad el algoritmo minimax con poda alfa-beta. En este apartado, el árbol de juego debe ser desplegado a medida que se va realizando la exploración, indicando claramente los pasos del algoritmo, numerando y etiquetando cada uno de ellos con la información relevante e indicando claramente los momentos de poda.
   IMPORTANTE: Únicamente se deben incluir y etiquetar los nodos descubiertos en la exploración del algoritmo. Puede que sea más cómodo dibujar el árbol en una hoja con formato apaisado, con el fin de tener suficiente espacio para el etiquetado.
   En caso de que se modifiquen las respuestas a las preguntas del apartado anterior, comentad el origen de las discrepancias. En caso de que sean las mismas, indicad la razón para ello.
      a. ¿Cuál es el valor minimax en la raíz del árbol de juego?
      b. ¿Cuál es la jugada óptima para A?
      c. ¿Cuál es la secuencia de jugadas óptimas?
      d. ¿Qué jugador gana la partida?

SOLUCIÓN:

(i) ¿Qué tipo de juego es?

a. Determinista / aleatorio
Determinista

b. Información completa / información no completa
Información completa

c. Suma cero / suma constante / suma variable
Suma cero

(ii) Para este tipo de juego, suponiendo que tanto A como B realizan elecciones óptimas ¿cuál es el algoritmo que determina cuál es la jugada óptima?
Minimax

(iii) Desplegad de manera completa el árbol de juego, indicando en cada nodo el estado del juego. Etiquetad los nodos terminales con el valor de la función de utilidad.
IMPORTANTE: Puede que sea más cómodo dibujar el árbol en una hoja con formato apaisado, con el fin de tener suficiente espacio para el etiquetado.
Árbol de juego en la siguiente página

(iv) Sobre el árbol desplegado, aplicad el algoritmo minimax, indicando claramente el valor minimax en cada uno de los nodos.

a. ¿Cuál es el valor minimax en la raíz del árbol de juego?
-1

b. ¿Cuál es la jugada óptima para A?
A juega 5

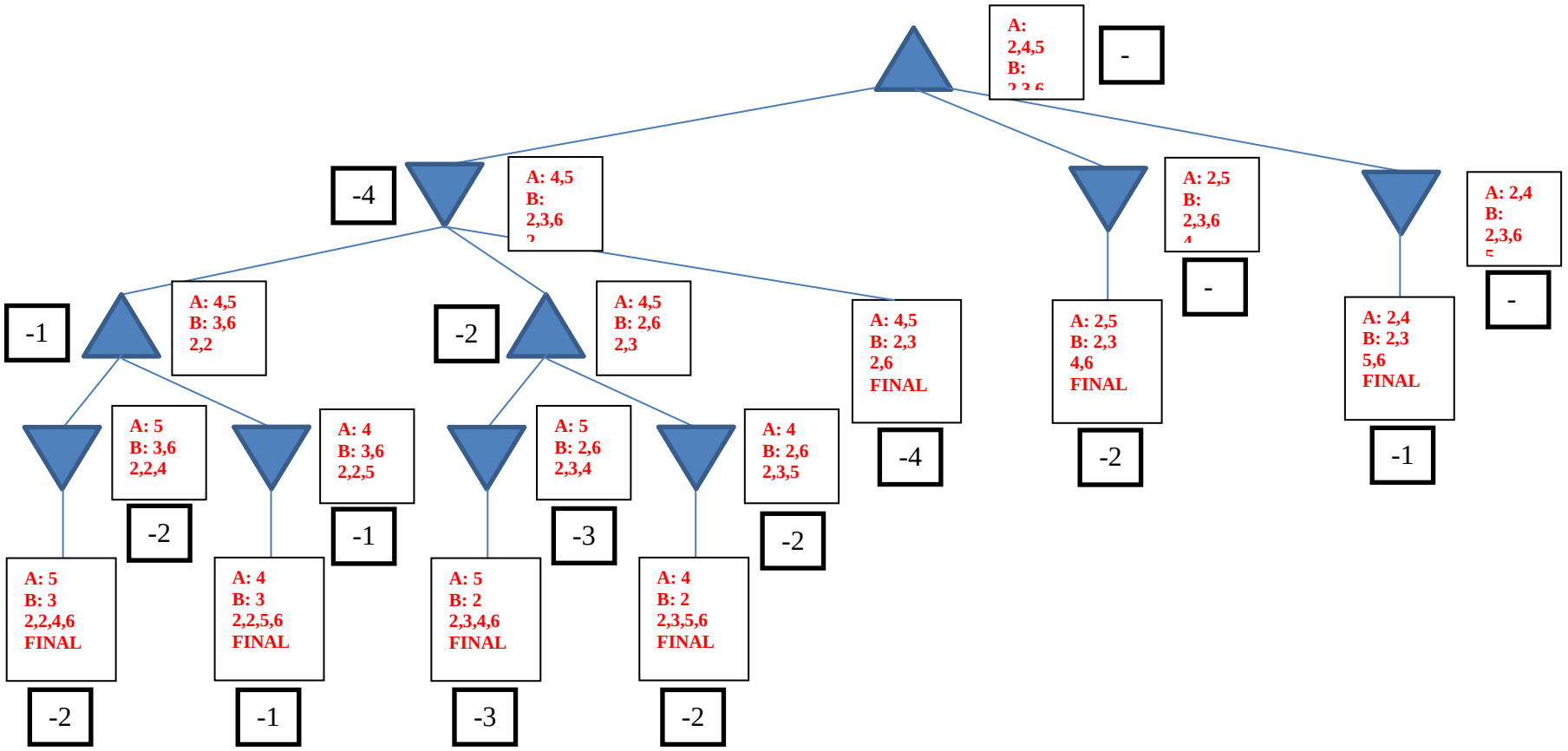
c. ¿Cuál es la secuencia de jugadas óptimas?
A juega 5
B juega 6

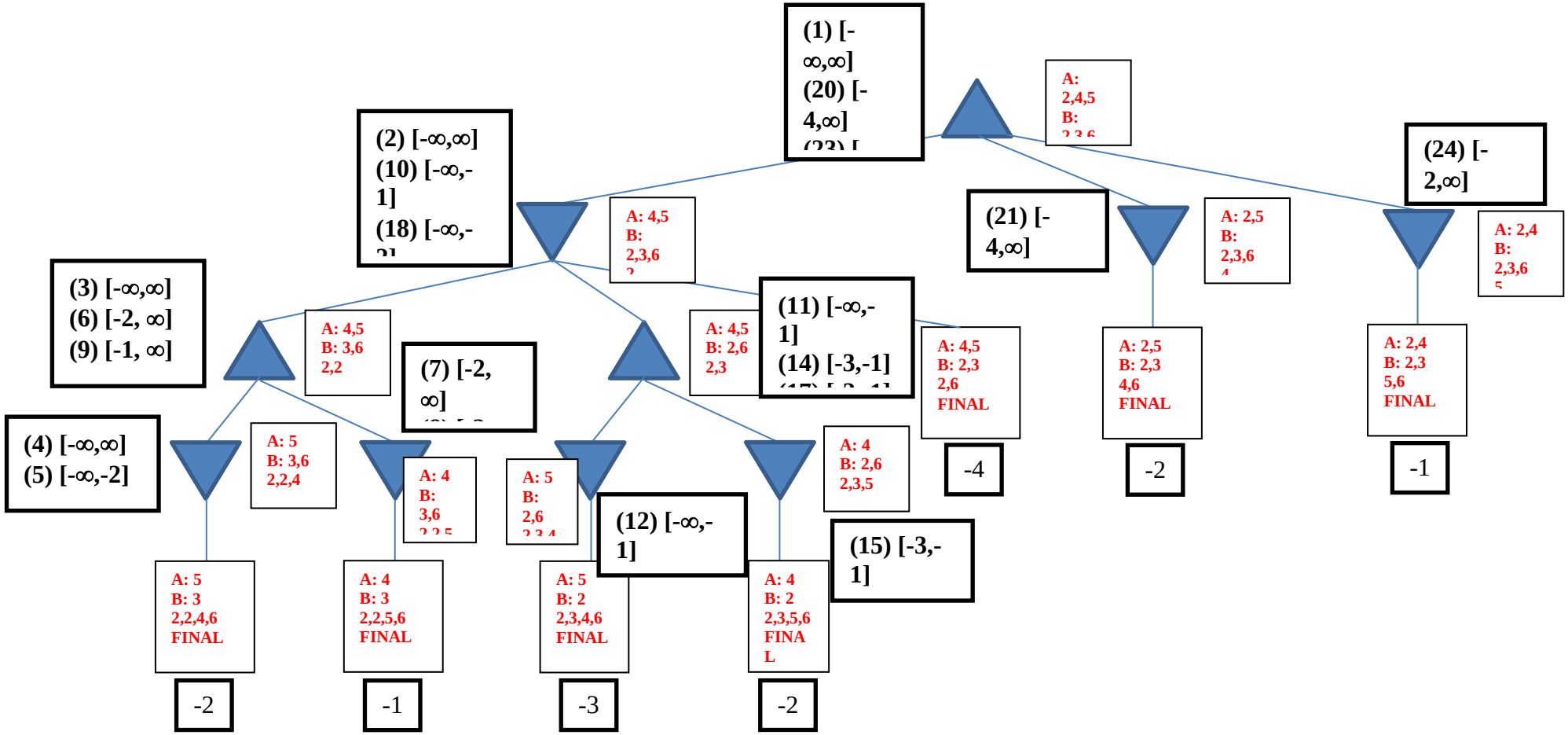
d. ¿Qué jugador gana la partida?
Gama B

(v) Aplicad el algoritmo minimax con poda alfa-beta. En este apartado, el árbol de juego debe ser desplegado a medida que se va realizando la exploración, indicando claramente los pasos del algoritmo, numerando y etiquetando cada uno de ellos con la información relevante e indicando claramente los momentos de poda.
IMPORTANTE: Únicamente se deben incluir y etiquetar los nodos descubiertos en la exploración del algoritmo. Puede que sea más cómodo dibujar el árbol en una hoja con formato apaisado, con el fin de tener suficiente espacio para el etiquetado.
En caso de que se modifiquen las respuestas a las preguntas del apartado anterior, comentad el origen de las discrepancias. En caso de que sean las mismas, indicad la razón para ello.

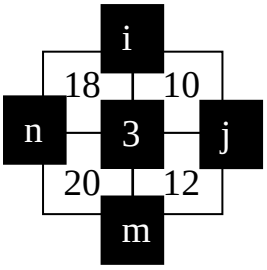
a. ¿Cuál es el valor minimax en la raíz del árbol de juego?
b. ¿Cuál es la jugada óptima para A?
c. ¿Cuál es la secuencia de acciones?
d. ¿Qué jugador gana la partida?

Las respuestas no cambian, dado que la poda alfa-beta permite identificar la estrategia minimax.


A:
2,4,5
B:
-
-4
A: 4,5
B:
2,3,6
A: 2,5
B:
2,3,6
-
A: 2,4
B:
2,3,6
-
-1
A: 4,5
B: 3,6
2,2
-2
A: 4,5
B: 2,6
2,3
A: 4,5
B: 2,3
2,6
FINAL
A: 2,5
B: 2,3
4,6
FINAL
A: 2,4
B: 2,3
5,6
FINAL
-4
-2
-1
A: 5
B: 3,6
2,2,4
-2
A: 4
B: 3,6
2,2,5
-1
A: 5
B: 2,6
2,3,4
-3
A: 4
B: 2,6
2,3,5
-2
A: 5
B: 3
2,2,4,6
FINAL
A: 4
B: 3
2,2,5,6
FINAL
A: 5
B: 2
2,3,4,6
FINAL
A: 4
B: 2
2,3,5,6
FINAL
-2
-1
-3
-2


(1) [-∞,∞]
(20) [-4,∞]
A: 2,4,5
B:
(24) [-2,∞]
A: 2,4
B: 2,3,6
5
A: 2,4
B: 2,3
5,6
FINAL
-1
(21) [-4,∞]
A: 2,5
B: 2,3,6
4
A: 2,5
B: 2,3
4,6
FINAL
-2
(2) [-∞,∞]
(10) [-∞,-1]
(18) [-∞,-
A: 4,5
B: 2,3,6
2
(11) [-∞,-1]
(14) [-3,-1]
A: 4,5
B: 2,6
2,3
A: 4,5
B: 2,3
2,6
FINAL
-4
(3) [-∞,∞]
(6) [-2, ∞]
(9) [-1, ∞]
A: 4,5
B: 3,6
2,2
(7) [-2, ∞]
(4) [-∞,∞]
(5) [-∞,-2]
A: 5
B: 3,6
2,2,4
A: 4
B: 3,6
2,2,5
A: 5
B: 3
2,2,4,6
FINAL
-2
A: 4
B: 3
2,2,5,6
FINAL
-1
A: 5
B: 2,6
2,3,4
A: 5
B: 2
2,3,4,6
FINAL
-3
(12) [-∞,-1]
A: 4
B: 2,6
2,3,5
(15) [-3,-1]
A: 4
B: 2
2,3,5,6
FINA
L
-2

7. Consideremos el siguiente puzle



El objetivo es utilizar A* para encontrar los valores enteros {i,j,m,n} entre 1 y 9 (ambos incluidos), de forma que la suma de los enteros en los cuadrados en negro adyacentes a un cuadrado que está en blanco sea correcta (por ejemplo: i+n+3 == 18). Cada entero solo puede ser utilizado solo una vez.

(i) Formalizad el problema:
   a. Estado inicial.
   b. Test que determina si se ha alcanzado el objetivo.
   c. Formalizad las acciones. En cada acción debe completarse la suma de uno de los cuadrados en blanco: La primera acción consiste en inicializar dos enteros, adyacentes a uno de los cuadrados en blanco (debéis elegir cuál). Las siguientes acciones consisten en inicializar un entero de forma que se complete la suma de otro cuadrado en blanco. El coste de una acción es la suma de los valores enteros utilizados en ese paso.

(ii) Definid una heurística monótona.

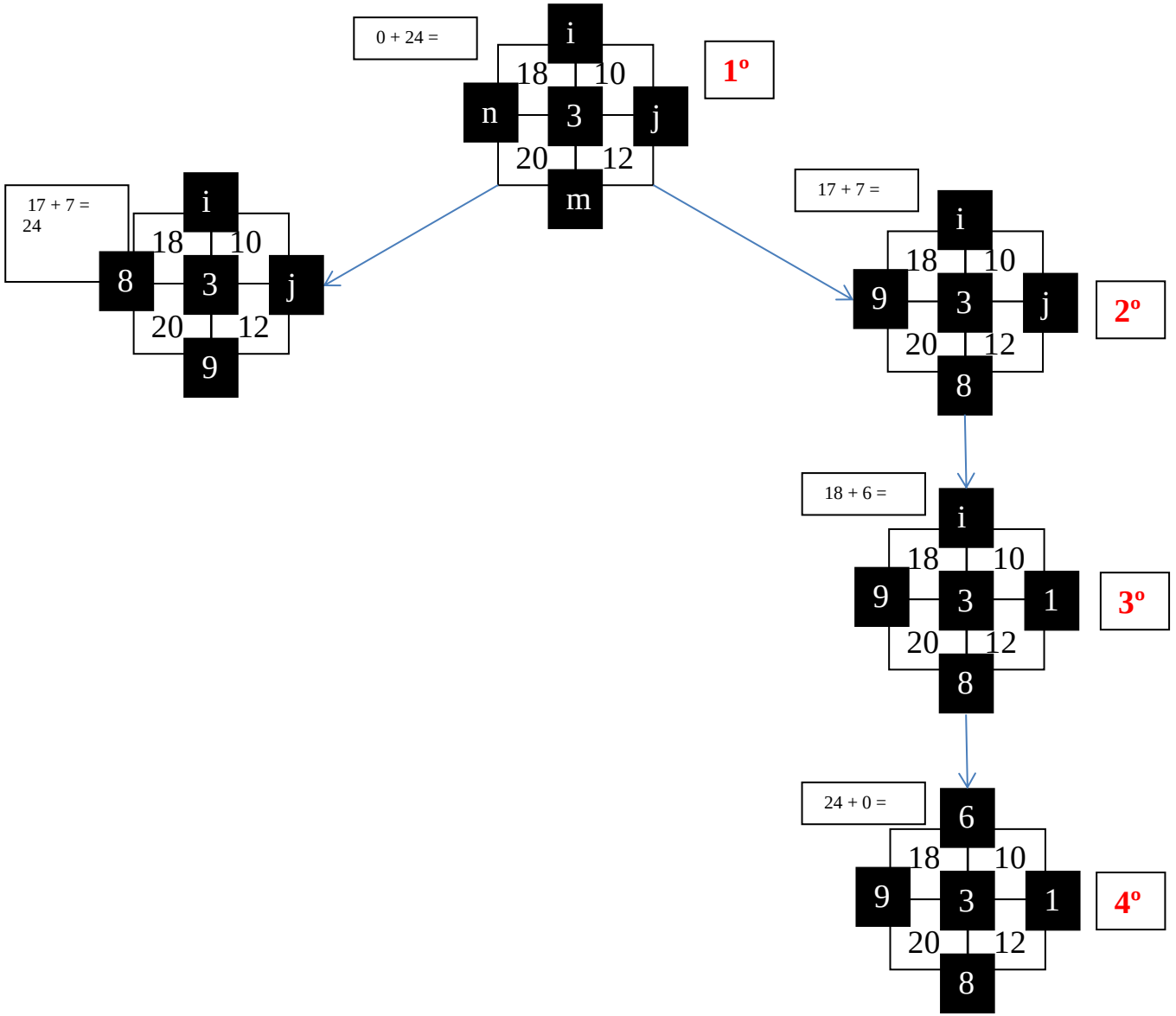
(iii) ¿Es la heurística definida admisible?

(iv) Encontrad una solución del puzle mediante A* con esta heurística y eliminación de estados repetidos. Para ello desplegad el árbol de búsqueda, etiquetando cada nodo n con los valores g(n) + h(n) = f(n). Indicad asimismo el orden de expansión de los nodos y qué nodos han sido eliminados por corresponder a estados repetidos.

(v) ¿Es la solución encontrada óptima? Justificad la respuesta.

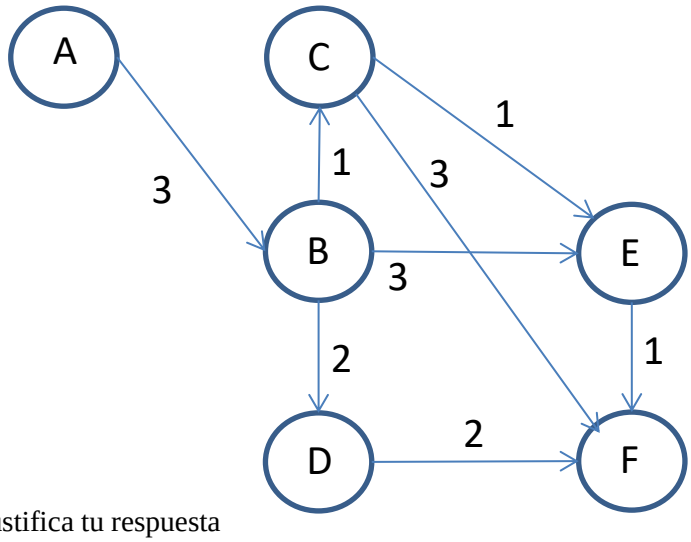
SOLUCIÓN:

Las acciones consisten en rellanar aquellas casillas con menor factor de ramificación posible.
Por ejemplo, la primera acción consiste en rellenar primero las dos casillas adyacentes al 20: solo hay dos posibilidades [n = 8, m = 9] o [n = 9, m = 8].
La heurística consiste en la suma de valores que faltan por rellenar:
Por ejemplo, en el estado inicial [h(inicial) = (20-3) + (10-3) = 24].
Es una heurística admisible (subestima la óptima) y monótona.
A* + eliminación de estados repetidos + heurística monótona garantiza encontrar la solución óptima.


0 + 24 =
i
18
10
n
3
j
20
12
m
1º
17 + 7 =
24
i
18
10
8
3
j
20
12
9
17 + 7 =
i
18
10
9
3
j
20
12
8
2º
18 + 6 =
i
18
10
9
3
1
20
12
8
3º
24 + 0 =
6
18
10
9
3
1
20
12
8
4º

8. Consideremos el siguiente grafo y la heurística para realizar búsqueda del camino de menor coste entre los nodos A y F.

N	h(n)
A	4
B	2
C	2
D	0
E	1
F	0



(i) ¿Es la heurística propuesta monótona? Justifica tu respuesta

La heurística es monótona, ya que cumple la desigualdad triangular para todas las aristas del grafo

Arista	h(n)		coste(n,n') + h(n')
AB	**4**	**≤**	**3 + 2 = 5**
BC	**2**	**≤**	**1 + 2 = 3**
BD	**2**	**≤**	**2 + 0 = 2**
BE	**2**	**≤**	**3 + 1 = 4**
CE	**2**	**≤**	**1 + 1 = 2**
CF	**2**	**≤**	**3 + 0 = 3**
DF	**0**	**≤**	**2 + 0 = 2**
EF	**1**	**≤**	**1 + 0 = 1**

(ii) ¿Es la heurística propuesta admisible? Justifica tu respuesta

Sí, porque toda heurística monótona es admisible

(iii) En este grafo, ¿garantiza A* con la heurística propuesta, sin eliminación de estados repetidos, encontrar la solución óptima?

Sí, porque la heurística monótona es admisible

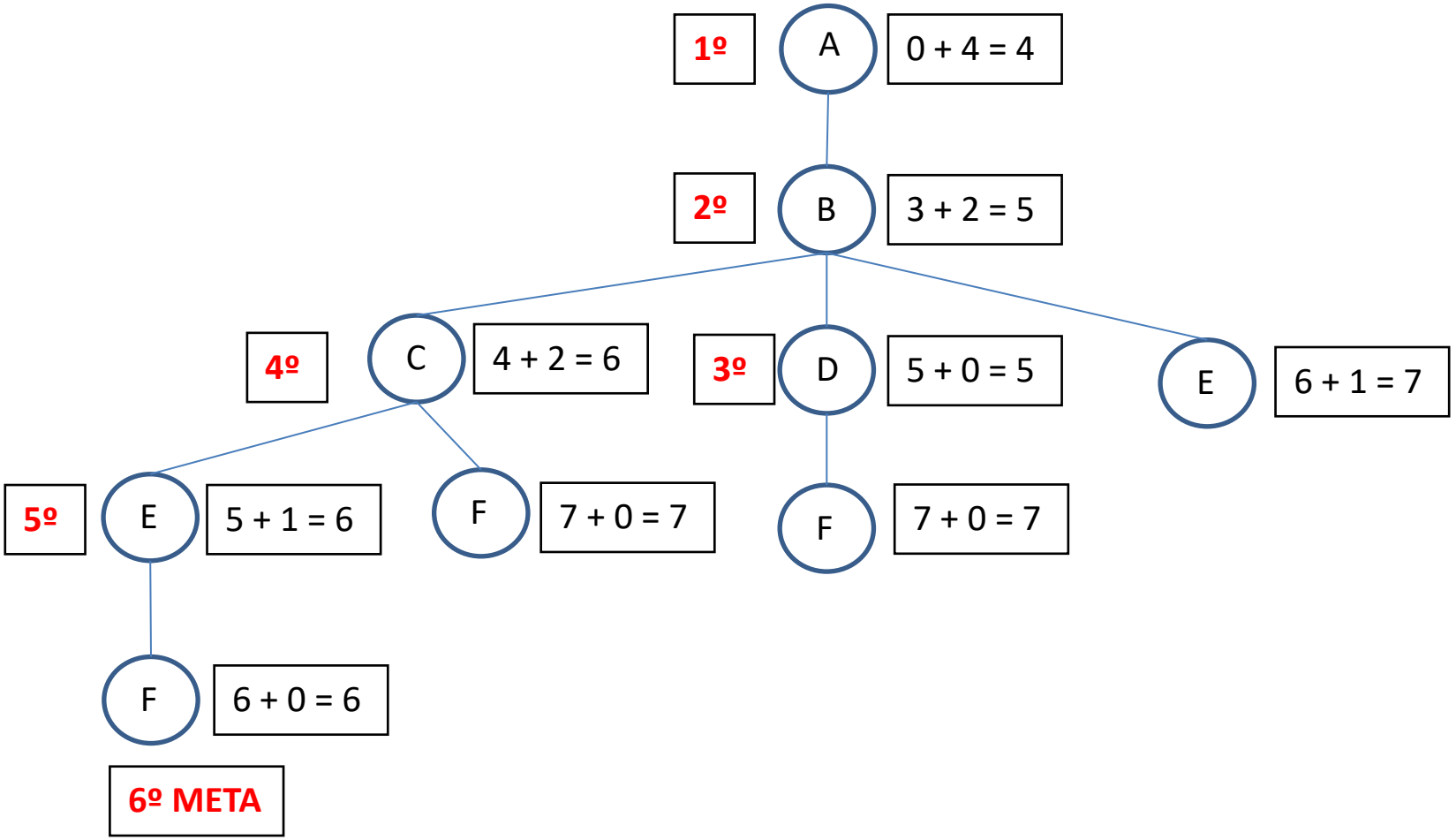
(iv) En este grafo, ¿garantiza A* con la heurística propuesta, con eliminación de estados repetidos, encontrar la solución óptima?
Sí, porque la heurística es monótona

(v) Encuentra una **solución mediante A* con la heurística propuesta y eliminación de estados repetidos**. Para ello, despliega el árbol de búsqueda, etiquetando cada nodo n con los valores g(n) + h(n) = f(n). Indicad asimismo el orden en el que se consideran los nodos para ser o bien expandidos o bien ser eliminados por corresponder a estados repetidos. En caso de empate, se expandirá primero el nodo más antiguo. En caso de que persista el empate, utiliza el orden alfabético para decidir el orden de exploración.

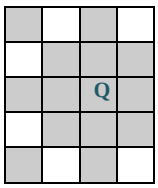
Árbol en hoja adjunta.

(vi) Indica cuál es la solución obtenida y el coste óptimo

A → B → C→ E→ F. Coste 6


1º
A
0 + 4 = 4
2º
B
3 + 2 = 5
4º
C
4 + 2 = 6
3º
D
5 + 0 = 5
E
6 + 1 = 7
5º
E
5 + 1 = 6
F
7 + 0 = 7
F
7 + 0 = 7
F
6 + 0 = 6
6º META

9. Consideremos un juego en el que dos jugadores, A y B, alternan turnos para colocar reinas sobre un tablero de ajedrez de 5 filas y 4 columnas.
   Una reina amenaza todas las casillas que estén en la fila, la columna y en las diagonales en la que se encuentra. Por ejemplo, todas las celdas sombreadas están amenazadas por la reina que se encuentra en este tablero



   En el juego el tablero inicialmente está vacío. El jugador A comienza el juego colocando una reina en alguna casilla de la columna más a la izquierda del tablero. El orden de exploración de las jugadas es de arriba a abajo. No es necesario explorar jugadas que, por simetría, conduzcan a posiciones equivalentes a otras ya generadas. El jugador B responde colocando una reina en alguna casilla de la siguiente columna que no esté amenazada por ninguna de las reinas que se encuentran en ese momento en el tablero. A la jugada de B responde A colocando una reina en alguna casilla de la columna siguiente no amenazada, y así sucesivamente.
   El juego termina cuando el jugador que en ese momento tiene el turno no puede colocar ninguna reina en una casilla no amenazada por las reinas ya colocadas en el tablero. El último jugador que ha podido colocar una reina en el tablero consigue 1 punto. Su oponente consigue 0 puntos.

   IMPORTANTE: Proporciona respuesta para cada uno de estos apartados

   (i) ¿Qué tipo de juego es?
      a. Determinista / aleatorio
      b. Información completa / información no completa
      c. Suma cero / suma constante / suma variable

   (ii) Para este tipo de juego, suponiendo que tanto A como B realizan elecciones óptimas ¿cuál es el algoritmo que determina cuál es la jugada óptima?

   (iii) Desplegad de manera completa el árbol de juego, indicando en cada nodo el estado del juego. Etiquetad los nodos terminales con el valor de la función de utilidad.
   IMPORTANTE: Puede que sea más cómodo dibujar el árbol en una hoja con formato apaisado, con el fin de tener suficiente espacio para el etiquetado.

   (iv) Aplicad el algoritmo minimax con poda alfa-beta. En este apartado se deben indicar claramente los pasos del algoritmo, numerando y etiquetando cada uno de ellos con la información relevante e indicando claramente los momentos de poda.
      a. ¿Cuál es la secuencia de jugadas óptimas determinada por el algoritmo?
      b. ¿Cuáles son los valores minimax en cada uno de los nodos del árbol de juego a lo largo de esta secuencia de jugadas óptimas?
      c. ¿Qué jugador gana la partida?

SOLUCIÓN:

(i) ¿Qué tipo de juego es?

a. Determinista / aleatorio

Determinista

b. Información completa / información no completa

Información completa

c. Suma cero / suma constante / suma variable

Suma constante

(ii) Para este tipo de juego, suponiendo que tanto A como B realizan elecciones óptimas ¿cuál es el algoritmo que determina cuál es la jugada óptima?

Minimax

(iii) Desplegad de manera completa el árbol de juego, indicando en cada nodo el estado del juego. Etiquetad los nodos terminales con el valor de la función de utilidad.

IMPORTANTE: Puede que sea más cómodo dibujar el árbol en una hoja con formato apaisado, con el fin de tener suficiente espacio para el etiquetado.

Árbol de juego en la siguiente página

(iv) Aplicad el algoritmo minimax con poda alfa-beta. En este apartado se deben indicar claramente los pasos del algoritmo, numerando y etiquetando cada uno de ellos con la información relevante e indicando claramente los momentos de poda.

a. ¿Cuál es la secuencia de jugadas óptimas?

La primera secuencia explorada completamente:

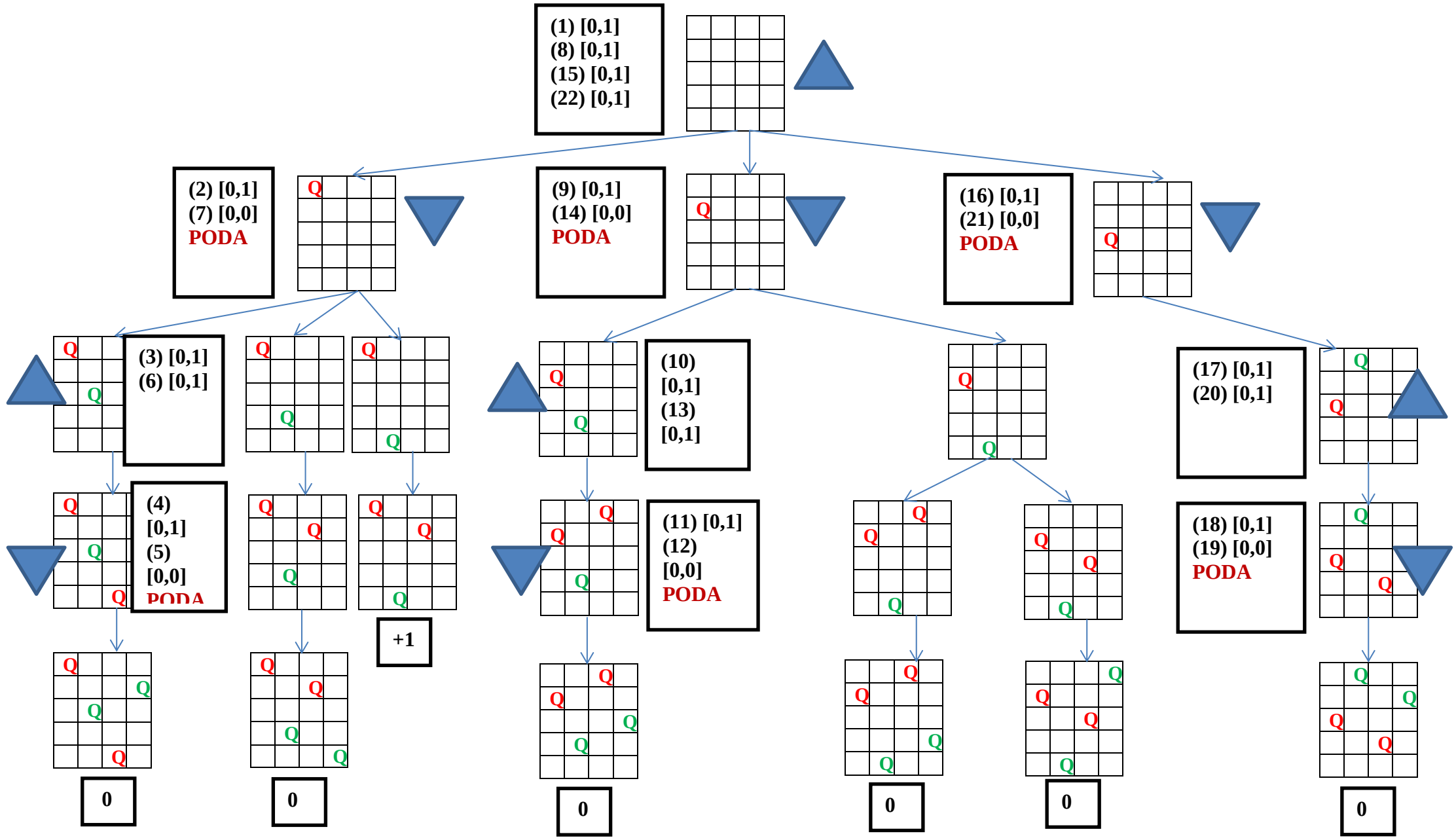
Q 1º fila → Q 3º fila → Q 5º fila → Q 2º fila

b. ¿Cuáles son los valores minimax en cada uno de los nodos del árbol de juego a lo largo de esta secuencia de jugadas óptimas?

El valor minimax en todos y cada uno de los nodos de esta secuencia de jugadas óptimas es 0.

c. ¿Qué jugador gana la partida?

Gana el jugador B


(1) [0,1]
(8) [0,1]
(15) [0,1]
(22) [0,1]
(2) [0,1]
(7) [0,0]
PODA
(9) [0,1]
(14) [0,0]
PODA
(16) [0,1]
(21) [0,0]
PODA
(3) [0,1]
(6) [0,1]
(4)
[0,1]
(5)
[0,0]
PODA
+1
(10)
[0,1]
(13)
[0,1]
(11) [0,1]
(12)
[0,0]
PODA
(17) [0,1]
(20) [0,1]
(18) [0,1]
(19) [0,0]
PODA
0
0
0
0
0