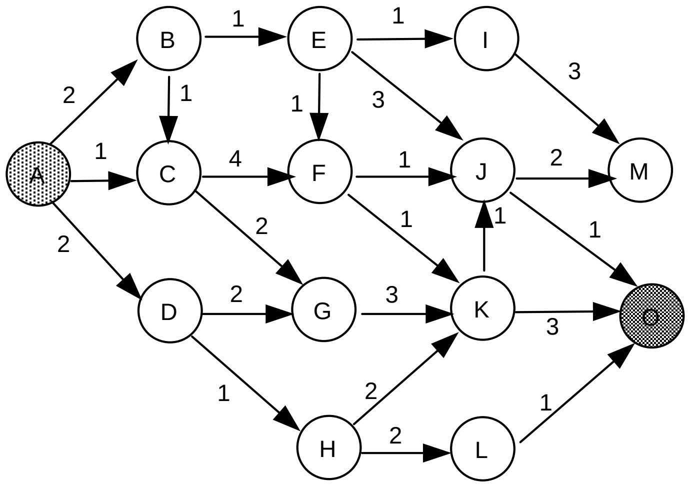
Dpto.de Sistemas Informáticos y Computación

SISTEMAS INTELIGENTES (GII)
EXAMEN 18-11-2013
ETSINF

NOTAS: La duración del examen es 1h 30 min

Pregunta 1 (5 puntos, Tiempo Estimado 40').

Sea el siguiente grafo donde cada arco indica su coste y la tabla indica la estimación del coste 'h' hasta la solución. El nodo 'A' es el estado inicial y el nodo 'O' es el estado final.



n	A	B	C	D	E	F	G	H	I	J	K	L	M	O
h(n)	6	5	6	6	3	5	5	4	8	2	2	1	5	0

1. (2.5 puntos) Muestra el árbol de búsqueda que resultaría de la aplicación de un algoritmo de tipo A (f(n)=g(n)+h(n)). Aplicar la versión grafo del algoritmo evitando nodos repetidos. Indica al final el número de nodos generados y expandidos.

    - Indica claramente el valor de la función de evaluación (f(n)) en cada nodo y el orden de expansión de los nodos. Si dos nodos tienen el mismo valor de f(n), expandir antes el nodo alfabéticamente anterior.

2. (1.5 puntos) De acuerdo a los datos del problema y el árbol desarrollado en el apartado anterior: ¿Devuelve el algoritmo la solución óptima? ¿La función heurística es admisible? ¿Y consistente (monótona)? Justifica todas las respuestas.

3. (1 punto) Sin desarrollar un árbol de búsqueda, contesta a las siguientes preguntas y justifica la respuesta:
    a. ¿Qué estrategia utilizarías si queremos encontrar la solución que atraviese el menor número de nodos? Indica una solución que encontraría esta estrategia.
    b. La aplicación de un algoritmo de profundización iterativa, ¿en qué nivel del árbol encontraría la solución?¿por qué?
    c. Si aplicamos un algoritmo en profundidad y no establecemos un límite máximo de profundidad, ¿encontraría el algoritmo una solución? ¿por qué?

Pregunta 2 (5 puntos, Tiempo Estimado 50').

En una terminal de contenedores se dispone de un conjunto de N pilas donde se apilan contenedores, entre los cuales algunos van a ser cargados en el próximo barco. Así, se distinguen entre dos tipos de contenedores: de tipo A, si se van a cargar en el próximo barco y de tipo B, en caso contrario. Dada una situación como la de la figura 1, el objetivo del SBR sería redistribuir los contenedores, de forma que los de tipo A queden liberados, es decir, no tengan ningún contenedor de tipo B encima (ver figura 2).

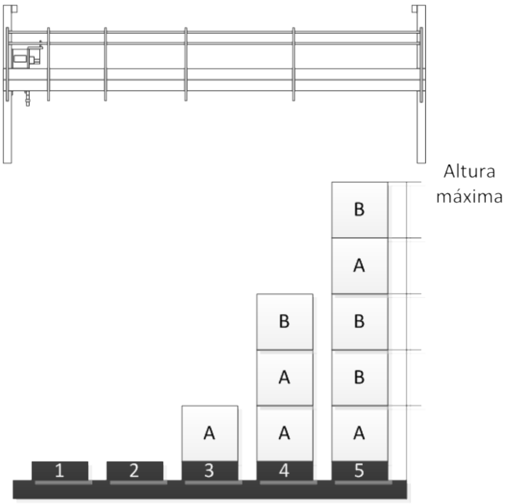


Figura 1. Situación inicial

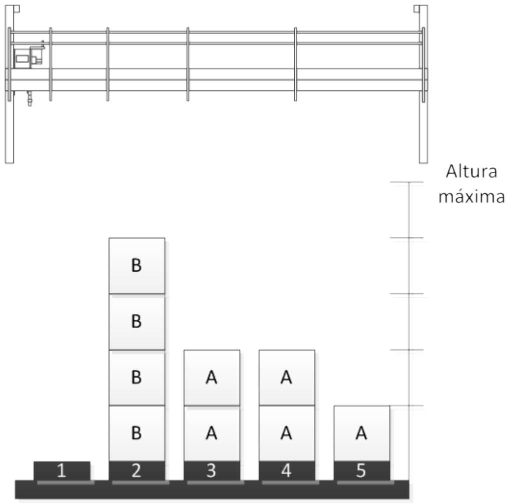


Figura 2. Una posible situación final

Los contenedores se pueden mover de una pila a otra por medio de una grúa que alcanza todas las pilas. Tanto el número de pilas como la altura de cada una de ellas están restringidos.

Asumamos que se utiliza el siguiente patrón para representar la información de un estado del problema:

(grua G^s [pila N^s contenedores C^s B^m]^m)

Donde:

- grua, pila y contenedores son símbolos constantes
- [pila N^s contenedores C^s B^m] se repite para cada una de las pilas
- G^s puede ser {libre, A, B}, es decir, se indica si la grúa está libre o qué tipo de contenedor sostiene
- N^s es el identificador de la pila, será un natural mayor que 0 que indicará la posición de la pila en la planta (1 pila del extremo izquierdo, n pila del extremo derecho, tal y como se indica en las figuras), las pilas estarán ordenadas de izquierda a derecha de menor a mayor identificador.
- C^s indica el número de contenedores en la pila
- B^m es la pila de contenedores, el primer contenedor de la lista es el que se encuentra en la parte superior de la pila

Además, se definen los siguientes hechos para indicar las restricciones sobre el número de pilas y sobre la altura máxima de las pilas.

(num-pilas X^s)   (max-altura Y^s)

Donde X^s e Y^s son valores enteros.

Se pide:

a) (1 punto) Escribir la base de hechos que corresponde a la situación inicial y final de las figuras 1 y 2. La representación de una pila vacía se deja a elección del alumno, teniendo en cuenta que el resto de reglas deben escribirse de acuerdo a esta representación.
b) (1.5 puntos) Escribir una regla para desapilar un contenedor de tipo B de una pila si éste bloquea un contenedor de tipo A inferior.
c) (1.5 puntos) Escribir una regla para apilar un contenedor de cualquier tipo en una pila que ya tenga al menos un contenedor. Se debe compensar la altura de las pilas, por lo que se podrá apilar un contenedor en una pila siempre y cuando la diferencia de altura entre dicha pila y la pila a su derecha sea inferior a 2 contenedores. Obviamente, esta condición no se aplica en la pila situada en el extremo derecho.
d) (1 punto) Se desea contabilizar el coste de todos los movimientos de contenedores realizados para alcanzar un determinado estado. El coste de desapilar cada contenedor de tipo A y de tipo B tiene un coste de 10 y 15, respectivamente. El coste de apilar cualquier tipo de contenedor es 5. Por ejemplo, la aplicación de estos costes daría un valor de 95 para el estado de la figura 2:
    a. Desapilar 4 contenedores B: 15*4 = 60
    b. Desapilar 1 contenedor A: 10*1 = 10
    c. Apilar los 5 contenedores: 5*5 = 25
    d. TOTAL = 95

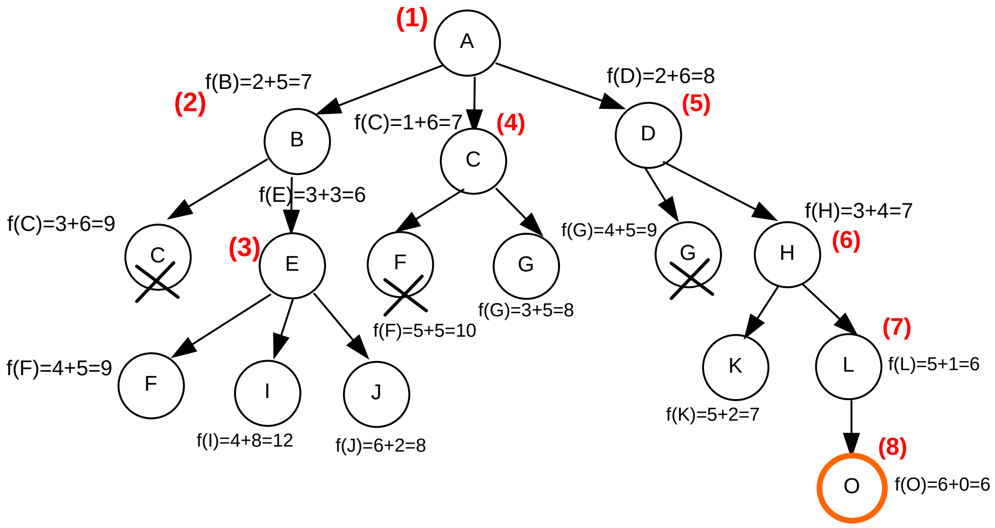
    Hacer las modificaciones necesarias en la Base de Hechos para poder representar esta información.

NOTAS:
- Estas reglas deben funcionar para trabajar con cualquier pila y con situaciones con cualquier número de pilas.
- Se pueden usar las siguientes funciones:
  (member$ <expression> <multifield-expression>)
  (length$ <multifield-expression>)
  (abs <numeric-expression>)

SOLUCIÓN

Pregunta 1

1) Se generan 16 nodos y se expanden 8.



2)

Sí, el algoritmo devuelve la solución óptima. Se puede ver en el grafo que no hay solución de menor coste.

No, la función no es admisible. Esto se puede observar en algunos casos, por ejemplo: h(H)=4, h*(H)=3; h(J)=2, h*(J)=1.

No, si la función no es admisible no puede ser consistente ya que la consistencia es una propiedad más fuerte que la admisibilidad. Esto se puede observar, por ejemplo, en: h(H)=4, h(L)=1, c(H,L)=2, por tanto no se cumple que h(H)<= h(L)+c(H,L). Esto también se puede observar entre el nodo B y E, D y H ó H y L, prueba de lo cual es que la función f(n) es decreciente en el árbol desarrollado.

3)

a) Para encontrar la solución más corta aplicaríamos la estrategia de anchura. Una solución que encontraría anchura es A-B-E-J-0, que es la solución más corta posible para este problema (4 pasos).

b) El algoritmo Profundización Iterativa encontraría la solución en el nivel 4 porque PI encuentra una solución de la misma calidad que anchura.

b) Sí, el algoritmo encontraría una solución porque el espacio de estados es finito y no contiene ciclos por tanto la búsqueda en profundidad no se quedaría estancada en un espacio de búsqueda infinito.

Pregunta 2

Apartado a)

```
(deffacts inicio
	(grua libre pila 1 contenedores 0 pila 2 contenedores 0 pila 3 contenedores 1 A pila 4
contenedores 3 B A A pila 5 contenedores 5 B A B B A)
	(max-altura 5)
	(num-pilas 5))
```

Situación final:

```
	(grua libre pila 1 contenedores 0 pila 2 contenedores 4 B B B B pila 3 contenedores 2 A A pila 4
contenedores 2 A A pila 5 contenedores 1 A)
```

Los hechos max-altura y num-pilas son estáticos y no varían.

Apartado b)

```
(defrule desapilar-B ;;se desapila un contenedor B si bloquea un contenedor A inferior
	(grua libre $?x pila ?p contenedores ?n B $?b $?y)
	(test (member A $?b))
	(test (= ?n (+ (length $?b) 1)))
=>
	(assert (grua B $?x pila ?p contenedores (- ?n 1) $?b $?y)))
```

Apartado c)

```
(defrule apilar
	(grua ?c $?x pila ?p contenedores ?n $?b pila ?p2 contenedores ?n2 $?b2 $?y)
	(max-altura ?max)
	(test (neq ?c libre))
	(test (> ?n 0))
	(test (= ?n (length $?b)))
	(test (= ?n2 (length $?b2)))
	(test (< ?n ?max))
	(test (< (abs (- ?n2 ?n)) 2))
=>
	(assert (grua libre $?x pila ?p contenedores (+ ?n 1) ?c $?b pila ?p2 contenedores ?n2 $?b2
$?y)))
```

Apartado d)

```
(deffacts inicio
	(grua libre pila 1 contenedores 0 pila 2 contenedores 0 pila 3 contenedores 1 A pila 4
contenedores 3 B A A pila 5 contenedores 5 B A B B A coste 0)
	(max-altura 5)
	(num-pilas 5)
	(coste desapilar A 10 B 15)
	(coste apilar A 5 B 5))
```