

Examen Febrero 2014 - Inteligencia Artificial

8 de febrero de 2014

1. Ejercicio (1 punto)

Completa el siguiente fragmento de código correspondiente a la función Expandir, dadas las especificaciones de las funciones para la formulación del problema Puzzle (C).

```
Lista expandir (tElemento *nodo)
    unsigned op;
    for(op=1; op<=NUM_OPERADORES; op++){
```



```
        InsertarUltimo (nuevo, sucesores);
    }//bloque 1
} //for
return sucesores;
} //expandir
```

```
puzzle.h
/* Comprueba si es posible aplicar un operador
a una configuracion determinada para el puzzle.
Devuelve 1 si el movimiento es valido y 0 en otro caso */

int esValido (unsigned op, tEstado *estado);

/* Aplica un operador a una configuracion concreta del puzzle.
devuelve la nueva configuracion concreta del tablero tras el movimiento */

tEstado *aplicaOperador (unsigned op, tEstado *estado);
```

```

busqueda.h
#ifndef _tElemento_
#define _tElemento_

typedef struct NodoBusqueda{
    tEstado *estado;
    struct NodoBusqueda *padre;
    unsigned operador;
    int costeCamino;
    int profundidad;
}tNodoBusqueda;

typedef tNodoBusqueda tElemento;

#endif

```

2. Ejercicio (2 puntos)

Formaliza mediante la lógica de predicados y demuestra mediante Resolución por Refutación que los **galos son feroces**.

Los romanos son temidos por su frialdad. Todo lo que es temido por su frialdad o por su fuerza es feroz. Los galos que viven en de Hispania y no son valientes. No todos los valientes son romanos. hay quienes viven en Hispania y odian a los que viven en Italia cuando se vuelven feroces.

3. Ejercicio (3,5 puntos)

Ante la necesidad de comprar un coche, hemos visitado un concesionario, donde nos han indicado las características de un monovolumen, el cual cumple con nuestras expectativas. El vendedor nos ha indicado que tiene capacidad para 7 personas: 2 asientos delanteros (piloto y copiloto), 3 en una fila central (dos en una sola pieza que ocupan las posiciones central e izquierda y uno en el lado derecho) y otros 2 asientos en una tercera fila.

- Todos los asientos pueden abatirse, excepto el del piloto.
- El asiento del copiloto podrá abatirse si y solo si el asiento justo de detrás también se encuentra abatido (es decir, este asiento de la segunda fila no es posible que este en posición normal si el del copiloto esta abatido).
- Los asientos de la segunda y tercera fila se pueden abatir cuando desee.
- Los asientos delanteros además, se pueden desplazar. Para que el asiento del piloto se pueda desplazar es necesario que los dos que se encuentran detrás (posición izquierda y central de la segunda fila) deben estar en posición normal.

- Para que el asiento del copiloto pueda desplazarse, el asiento de detrás (derecha de la segunda fila) debe estar en posición normal.
- Un asiento abatido no se puede desplazar.
- Un asiento desplazado no se puede abatir.

Obviamente, todas las operaciones de abatir y desplazar sólo se pueden llevar a cabo si los asientos se encuentran en posición normal. Y además, son operaciones individuales, es decir no se pueden abatir y desplazar dos asientos simultáneamente. La operación abatir tiene un coste igual a 2, al igual que la operación desplazamiento tiene un coste igual a 1.

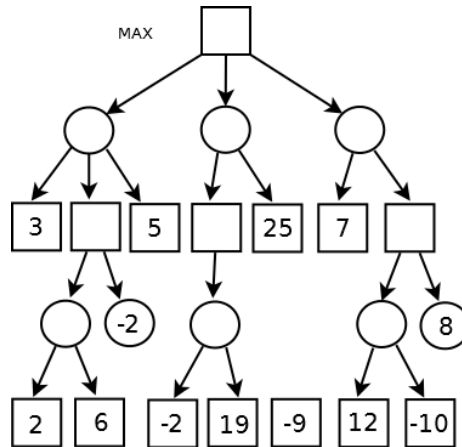
Nos indica el vendedor que el coche nos lo entregan con los asientos de la primera fila desplazados, los de la segunda en posición normal y los de la tercera fila abatidos. Sin embargo nuestro coche dispone de un sistema inteligente que es capaz de configurar automáticamente las posiciones del vehículo para que se adapte a nuestras necesidades. Concretamente, para poder llevarnos el coche tenemos que tener en cuenta que seremos 2 adultos (piloto y copiloto) y 2 niños (sentados en la tercera fila) y la fila intermedia estará abatida para poder llevar las maletas.

Realiza la formalización como un problema cuya resolución puede realizarse mediante búsqueda en espacio de estados. Tal y como se ha estudiado en la asignatura.

- A. Describe el tipo de datos `tEstado`, indicando estado inicial y final.
- B. Describe los operadores.
- C. Diseña las funciones `testObjetivo`, `esValido` y `aplicaOperador`.
- D. Diseña la función `coste`, y calcula el coste de cualquier operador.
- E. Define una función heurística, `e` indica si es admisible o no lo es.

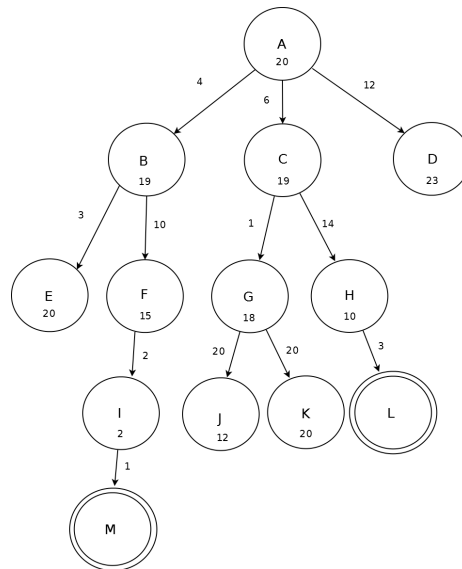
4. Ejercicio (1,5 puntos)

MiniMax y poda alfa - beta.¹



5. Ejercicio (1 punto)

Algoritmo A* y calcular el coste real.



¹Aquí ya solo copie lo imprescindible, como comprenderéis ya estaba un poco harto