Lógica para la Computación

Convocatoria de junio, 18/01/23

Nombre: DNI: Plan antiguo □

NOTA: Es necesario un mínimo de 3 ptos (el 50% de la puntuación total) en la prueba para sumar las prácticas correspondientes. La duración del examen es de 2 horas.

1. (1.5 ptos) Sean los predicados `humano(I)` y `madre(P,M)`, ciertos sii un individuo `I` es humano y `M` es la madre de `P`, respectivamente. Consideremos como implementacin el siguiente código PROLOG:

```
humano(elena).
humano(X) :- madre(X,Y), humano(Y).
madre(juan,elena).
```

   y la pregunta `:- humano(X).`

   Supongamos además que el algoritmo de resolución SLD de PROLOG implementase una exploración ascendente de las cláusulas del programa, y un tratamiento FIFO de las resolventes:

   (a) (0.75 ptos) Indicar la razón por la que no sería posible obtener una respuesta. Justificar la respuesta.

   No obtendríamos respuesta, la resolución entraría en un ciclo de resolución. Para justificarlo bastaría, por ejemplo, con incluir y comentar adecuadamente un árbol de resolución para una pregunta del tipo `:- humano(X)`.

   (b) (0.75 ptos) Indicar tres formas de modificar el programa para que la respuesta sea posible.

   Para evitarlo podríamos, por ejemplo:

   i. Reescribir la segunda cláusula de humano/2 como

```
humano(X) :- madre(X,Y), freeze(Y, humano(Y)).
```

   ii. Reescribir la segunda cláusula de humano/2 como

```
humano(X) :- madre(X,Y), when(nonvar(Y), humano(Y)).
```

   iii. Reescribir el predicado humano/2 como

```
humano(elena).
humano(X) :- humano(Y), madre(X,Y).
```

   iv. Reescribir el predicado humano/2 como

```
humano(X) :- madre(X,Y), humano(Y).
humano(elena).
```

   v. Reescribir el predicado humano/2 como

```
humano(X) :- humano(Y), madre(X,Y).
humano(elena).
```

   NOTA: Obviamente no se admitirán "variaciones" equivalentes entre sí como distintas. Por ejemplo, no hay diferencia operativa alguna entre el programa:

```
humano(X) :- humano(Y), madre(X,Y).
humano(elena).
madre(juan,elena).
```

   y el programa:

```
madre(juan,elena).
humano(X) :- humano(Y), madre(X,Y).
humano(elena).
```

2. (1.5 ptos) Dado el programa PROLOG definido por la cláusula:

```
adivina(L1, L2) :- append(_,L3,L2), append(L1,_,L3).
```

describir la semántica declarativa del predicado `adivina/2`. Justificar la respuesta.

El predicado `adivina/2` *se verifica sii la lista* `L1` *es una sublista de la lista* `L2`[1]*. Basta tener en cuenta que:*

```
L2 = _ + L3
L3 = L1 + _
```

de donde:

```
L2 = _ + L3 = _ + L1 + _
```

Esto es, `L2` *se obtiene añadiendo una lista (cualquiera) por delante de* `L1` *y otra lista (cualquiera) por detrás de* `L1`*, de donde se extrae trivialmente la afirmación inicial. Obviamente los argumentos* `L1` *y* `L2` *han de ser listas, ya que así lo requiere el predicado predefinido* `append/3`*.*

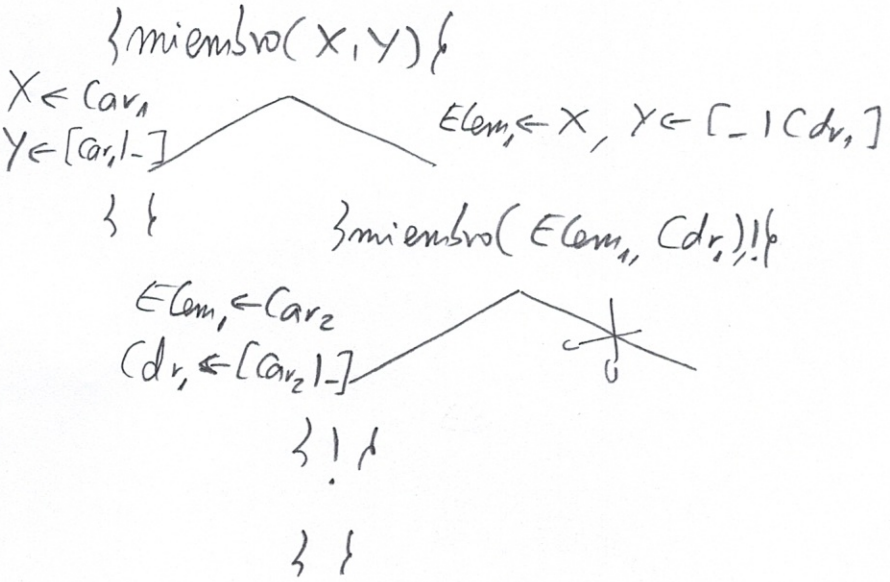
[1] Manteniéndose por tanto el orden de los elementos de `L1` en `L2` y también las relaciones de adyacencia entre aquellos.

3. (1.5 ptos) Consideremos el código PROLOG siguiente:

```
miembro(Car,[Car|_]).
miembro(Elem, [_|Cdr]) :- miembro(Elem,Cdr), !.
```

Explicar y justificar el efecto del corte en este caso.

El efecto es que, de entrar la resolución en la regla recursiva, si el primer objetivo de la cola se resuelve de forma exitosa, no se buscan más soluciones ... aunque las hubiera. Basta ver el árbol de resolución para la pregunta `:- miembro(X,Y).`



Dependiendo del tipo de pregunta que se haga, pudiera incluso no haber ningún efecto, al situarse el corte al final de la última cláusula del predicado, si bien ello no representa el caso general. De plantearse así la respuesta del alumno, se evaluará en función de la justificación facilitada, aunque nunca obtendrá la nota máxima.

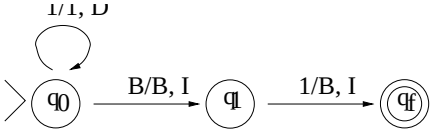
4. (1.5 ptos) Describir una Máquina de Turing, exclusivamente mediante un grafo de estados, que obtenga el predecesor de un número según la codificación binaria que representa a un número decimal n mediante una secuencia de $n+1$ 1's.

Ejemplos:	0	$\rightarrow$	cadena vacía
	1	$\rightarrow$	1
	2	$\rightarrow$	11
	3	$\rightarrow$	111
	...		

De este modo, a una entrada $1 \overbrace{\ldots}^{n+1)} 1$ corresponderá una salida $1 \overbrace{\ldots}^{n)} 1$.

Justificar la respuesta y trazar sus movimientos para la entrada $w = 111$.

Uno de los posibles grafos de estados sería el siguiente:



El estado inicial q_0 simplemente consume la entrada avanzando en ella mientras haya 1's (si encuentra otro término que no sea un 1 o un B, se para en error). Detectado el final de la entrada mediante acceso a un B, la TM pasa al estado q_1, que simplemente retrocede una posición para acceder al último 1 de la entrada y borrarlo para luego llegar al estado final q_f y aceptar la entrada.

En cuanto a la traza correspondiente a la entrada $w = 111$, sería en este caso particular:

$$q_0111 \vdash 1q_011 \vdash 11q_01 \vdash 111q_0B \vdash 11q_11 \vdash 1q_f1$$