Evaluación (1ª instancia)

Duración: 3 horas

Ejercicio 1 [40 puntos]

Considere un tablero de N x N celdas en donde se intentarán ubicar palabras de modo que el tablero quede completo -todas sus celdas están ocupadas por una letra- y válido -todas las palabras formadas pertenecen a un diccionario-. La figura que sigue muestra un tablero completo y válido de tamaño 3 X 3, se asume que el diccionario incluye las palabras: *los, asa, sol, las, oso, sal*.

l	o	s
a	s	a
s	o	l

Un tablero N X N se representa mediante una lista de N filas. Las filas, a su vez, son listas de largo N.

Las palabras del diccionario están representadas por medio del predicado `palabra\1` que tiene como argumento una lista de letras:

```
palabra([l,o,s]).
palabra([s,o,l]).
palabra([a,r,e,n,a]).
palabra([p,l,a,n]).
...
```

Implemente en Prolog los predicados que se describen a continuación:

a) `pal_valida_en_fila(+T)<-`
T tiene al menos una palabra válida en alguna de sus filas.

b) `col_j(?T, +J, -Col) <-`
Col es la columna j-ésima del tablero T (representado como lista de filas).

c) `tablero_completo_valido(?T, +N) <-`
El tablero T (representado como lista de filas), de dimensión N X N, es completo y válido según lo definido anteriormente.

d) `todos_tableros_sin_rep(?T, +N, -TT) <-`
TT es la lista de todos los tableros de dimensión N X N completos y válidos, compatibles con el tablero T, que no contienen palabras repetidas.

Observación importante
En las partes b), c) y d) se asume que:
El argumento T es una lista que contiene N listas de largo N.
El argumento T puede estar:

- `completamente instanciado: [[l, o, s],[a, s, a],[s, o, l]]`
- `completamente sin instanciar: [[A, B, C],[D, E, F],[G, H, I]]`
- `semi-instanciado: [[A, B, s],[o, E, F],[G, H, I]]`

Ejercicio 2 [5 puntos]

Indique si las siguientes afirmaciones son verdaderas o falsas:

a) Si una fórmula **f** es consecuencia lógica de un conjunto de fórmulas **S**, entonces se cumple que **S U {f}** es insatisfactible.

b) Un objetivo definido no puede contener ningún literal positivo.

c) Las cláusulas de Horn contienen al menos un literal positivo.

d) Un m.g.u. es una sustitución con un solo elemento.

e) Un árbol SLD contiene todas las respuestas correctas para una consulta.

Ejercicio 3 [30 puntos]

Dado el siguiente conjunto de cláusulas para los predicados **p** y **q**:

```
C1: p(X,Y) :- q(Y,X).
C2: p(X,Y) :- p(Y,X).
C3: p(X,Y) :- q(X,Y), XN is X+1, p(Y,XN).
C4: q(1,2).
C5: q(1,0).
C6: q(3,1).
```

a) Dibuje el árbol SLD correspondiente a la consulta **?- p(1,V)**, suponiendo que la regla de computación toma el átomo de más a la izquierda para la resolución. Indique para cada rama de éxito cuál es la respuesta computada correspondiente.

b) ¿Qué respuestas dará un intérprete prolog que aplique la regla de computación de la parte a), tomando las cláusulas en su orden de aparición y haciendo recorrida en profundidad del árbol?

c) Diga cuáles serían las respuestas de la parte b) si las cláusulas de **p** estuvieran en el siguiente orden: `C3, C2, C1.` Justifique su respuesta.

d) ¿Qué respuestas daría un intérprete prolog como el de la parte b) si se eliminara la clásula C2? Justifique su respuesta.

e) Considere las siguientes variantes para la cláusula C1:

i. `C1: p(X,Y) :- !, q(Y,X).`

ii. `C1: p(X,Y) :- q(Y,X), !.`

¿Qué respuestas daría un intérprete prolog como el de la parte b) para la consulta **?- p(1,2)**, para cada una de las variantes de C1?

Ejercicio 4 [10 puntos]

Escribir el siguiente predicado:

`cumple_nocumple(+Prop, +L, ?L1, ?L2) <-`
La lista L1 contiene todos los elementos de L que cumplen la propiedad Prop y la lista L2 contiene todos los elementos de L que no la cumplen.

Observaciones
- No se debe probar dos veces la propiedad `Prop` p ara un mismo elemento de la lista.
- `Prop` es el nombre de algún predicado que se asume definido previamente. Por ejemplo, si se cuenta con el predicado `par\1`, que verifica si un número es par, la siguiente consulta es verdadera:
  `?- cumple_nocumple(par, [2,3,5,6,1,11,10], [2,6,10], [3,5,1,11]).`