

Lógica para la Computación

Convocatoria de julio, 23/06/25

Nombre:

DNI:

PCEO ☐

Grupo A ☐

Grupo B ☐

NOTA: Es necesario acumular un mínimo de 3 pts (el 50% de la puntuación total) en esta prueba para sumar las notas prácticas. La duración del examen es de 2 horas, y se puntúa sobre 6 pts.

1. (1 pts) Consideremos el predicado definido mediante la fórmula siguiente:

$$\forall X, Y \text{ [} \forall P [P(X) \Leftrightarrow P(Y)] \Rightarrow [X = Y] \text{]}$$

conocido como *principio del tercero excluido*. Responder a las siguientes cuestiones:

- (a) (0.5 pts) Describir su semántica declarativa.

Si para dos elementos X e Y es cierto que una propiedad cualquiera P se verifica en uno sii se verifica en el otro, entonces ambos elementos son el mismo.

- (b) (0.5 pts) Razonar si es posible su tratamiento en base al algoritmo de resolución SLD.

No es posible, ya que los funtores utilizados en el algoritmo SLD descrito son símbolos constantes, cuando la implementación del axioma de inducción requeriría el recurso a funtores variables.

2. (1 pts) Dado el predicado PROLOG definido por las cláusulas siguientes:

```
p(X,X) :- !, fail.  
p(X,Y).
```

- (a) (0.5 pts) Describir la semántica declarativa de cada una de las cláusulas.

La primera cláusula falla sii sus dos argumentos unifican. La segunda, alternativa exclusiva de la primera, siempre se verifica. Esto es, el predicado $p/2$ identifica pares de argumentos no unificables.

- (b) (0.5 pts) Proponer una implementación alternativa, sobre una única cláusula.

Una alternativa posible es la dada por la cláusula:

```
p(X,Y) :- (X = Y, !, fail ; true).
```

3. (1 pts) Implementar un predicado `assertb/1` que haga sensible `asserta/1` al retroceso. Esto es, si en el transcurso de una resolución se produce un retroceso sobre el objetivo `asserta(Clausula)` que había determinado la inclusión de `Clausula` en nuestra base de datos, el intérprete deberá ahora eliminarla. Justificar la respuesta, describiendo la semántica declarativa del nuevo predicado `assertb/1`.

Ver el Ejemplo 12.3.1 en el fichero

*Documentos y Enlaces/Material de Estudio/Prolog/prolog.pdf
de la entrada Moovi de la asignatura, y que ha servido de guía a las clases.*

4. (1.5 pts) Completar el programa de cláusulas PROLOG siguiente:

```
pegar [] a L genera L.
pegar [Car|Cdr1] a L genera [Car|Cdr3] :- pegar Cdr1 a L genera Cdr3.
```

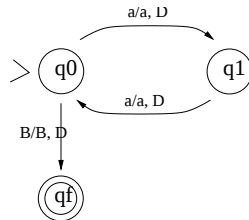
de forma que podamos plantear preguntas del tipo :- pegar L1 a L2 genera L. Justificar la respuesta.

Bastará con añadir las cláusulas siguientes:

```
:- op(100,xfx,a).           :- op(200,xfx,genera).           :- op(300,fx,pegar).
```

todos ellos como no asociativos, dado que la sintaxis del predicado no lo requiere al no repetirse ninguno de ellos en ninguna de las cláusulas. Los dos primeros han de ser binarios, **a** tomando como argumentos al primero y segundo del predicado, y **genera** tomando como argumentos al sintagma definido por **a** y al tercer argumento del predicado. En cuanto a **pegar**, toma como único argumento al sintagma definido por **genera**. Sus prioridades son, por tanto, las acordes con la distribución de argumentos comentada.

5. (1.5 pts) Dado un alfabeto $\Sigma = \{a\}$ y la Máquina de Turing asociada al grafo de estados siguiente:

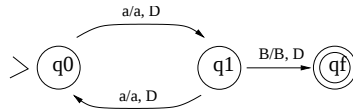


(a) (.75 pts) Identificar el lenguaje $\mathcal{L}$ reconocido por la máquina. Justificar la respuesta.

El lenguaje es $\mathcal{L} = \{a^{2n}, n \geq 0\}$, basta para ello interpretar la arquitectura del grafo. Las transiciones entre los estados q_0 y q_1 aseguran en cada ciclo el reconocimiento de un par de **a**'s. Dado que para acceder al estado final q_f hay que partir del inicial q_0 , ello implica que las cadenas reconocidas son bien la cadena vacía, bien secuencias no vacías de **a**'s con longitud par.

(b) (.75 pts) Modificar el grafo para que reconozca el complementario $\Sigma^* \setminus \mathcal{L}$. Justificar la respuesta.

Una posible solución es la dada por el grafo siguiente:



Basta tener en cuenta que en esta nueva configuración solo es posible alcanzar el estado final q_f a partir de q_1 , lo que supone que previamente se ha completado una transición a partir del inicial q_0 precedida o no de un ciclo sobre q_1 . En cualquiera de los casos, al alcanzar q_f habremos reconocido una secuencia de **a**'s de longitud impar.