Curso 2012 – 2013
Convocatoria de Diciembre

Lenguajes de Programación. (Examen sobre 6 puntos, necesarios 3 puntos para aprobar).

1. (1.5 puntos) Describir una Máquina de Turing capaz de reconocer el lenguaje L = { $0^n 1^n$, n >= 1 }.

Una **Máquina de Turing** es un dispositivo teórico que manipula símbolos en una cinta infinita de acuerdo a una tabla de reglas. Esta máquina teórica puede adaptarse para simular cualquier lógica computacional aunque está especializada en lo que se conoce como paradigma imperativo.

La Máquina de Turing se define como:

$$M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$$

Donde:

- **Q** es el **conjunto** finito **de estados** de la unidad de control.
- ∑ es el **conjunto** finito **de símbolos** de entrada.
- **Ґ** es el **conjunto** completo **de símbolos** de cinta /∑**cҐ** (contenido)
- ʒ es la **función de transición**. Los argumentos de ʒ(q,X) son: un estado q y un símbolo de la cinta X (0 o 1). De estar definido (p,Y,D) p sería el siguiente estado, Y un símbolo de **Ґ** y D la dirección (izquierda o derecha).
- **q_0** es el **estado inicial** y por lo tanto pertenece a **Q**.
- **B** es el símbolo de espacio en blanco y por lo tanto pertenece a **Ґ.**
- **F** es el conjunto final o de aceptación.

Un diseño para una Máquina de Turing que acepte el lenguaje L = { $0^n 1^n$, n >= 1 } sería:

M = ({q_0, q_1, q_2, q_3, q_4}, {0, 1}, {0, 1, X, Y, B}, ʒ, q_0, B, {q_4})

Donde:

- Los estados posibles (Q) son: **{q_0, q_1, q_2, q_3, q_4}**
- El conjunto de símbolos de entrada (∑) es: **{0, 1}**
- El conjunto de símbolos posibles (**Ґ**) sería: **{0, 1, X, Y, B}**
- La función de abstracción (ʒ) sería: **ʒ**
- El estado inicial: **q_0**
- **B** representaría el carácter de espacio en blanco**.**
- El conjunto de estados finales (aceptación o **F**) sería: **{q_4}**

Tabla que especifica el comportamiento (conjunto de reglas) sería:

Estado	0	1	X	Y	B
q_0	(q_1, X, D)	-	-	(q_3, Y, D)	-
q_1	$(q_1, 0, D)$	(q_2, Y, I)	-	(q_1, Y, D)	-
q_2	$(q_2, 0, I)$	-	(q_0, X, D)	(q_2, Y, I)	-
q_3	-	-	-	(q_3, Y, D)	(q_4, B, D)
q_4	-	-	-	-	-

La secuencia de movimientos para la entrada 0011 (aceptada por la MT) sería:

$q_0 0011 \vdash X q_1 011 \vdash X0 q_1 11 \vdash X q_2 0Y1 \vdash q_2 X0Y1 \vdash$
$X q_0 0Y1 \vdash XX q_1 Y1 \vdash XXXY q_1 1 \vdash XX q_2 YY \vdash X q_2 XYY \vdash$
$XX q_0 YY \vdash XXY q_3 Y \vdash XXXYY q_3 B \vdash XXYYB q_4 B$

2. (1.5 puntos) Describir las reglas de inferencia que definen la β-reducción, comentando brevemente su significado.

Denominamos **β-reducción** a un proceso de derivación que sirve de base a la interpretación funcional.

Sean M y N dos $T_\Sigma(x)$ (conjunto completo de combinaciones posibles dentro del lenguaje) decimos que M se β-reduce a N si existe una derivación que transforme M en N mediante la combinación de las siguientes reglas de inferencia:

$$(red) : \frac{}{(\lambda x P)Q \rhd P[x := Q]}$$

$$(\lambda) : \frac{M \rhd M'}{\lambda x M \rhd \lambda x M'} \qquad (1) : \frac{M \rhd M'}{MN \rhd M'N} \qquad (2) : \frac{M \rhd M'}{NM \rhd NM'}$$

4

3. (1.5 puntos) Implementar un predicado en PROLOG de sintaxis duplicar(Lista, Resultado) tal que Resultado es la lista resultante de duplicar en la lista cada uno de sus elementos.

Ejemplo: La respuesta a la pregunta duplicar([a, b, c, d],X) es X = [a, a, b, b, c, c, d, d].

duplica([], L).

duplica([Car|Cdr],[Car,Car|L]) :- duplica(Cdr,L).

4. (1.5 puntos) Explicar cuál es el origen de los ciclos de unificación en PROLOG, ilustrándolo con un ejemplo.

Podemos definir la **unificación** como el paso de variables a constantes mediante un proceso de resolución, es decir, una unificación es un proceso de sustitución que aplicado a dos términos los iguala bien porque sean idénticos o porque sus variables pueden instanciarse por "objetos" te manera que sean equivalentes.

Ejemplo:

fecha(D,M,1998) = fecha(D1,mayo,A)

En PROLOG el origen de los ciclos de unificación viene definido por el Algoritmo de Unificación de Robinson que determina que dos funciones son unificables si:

1.- Las funciones tiene en común la letra o palabra que usen para definirse, es decir, su identificador.
2.- Las funciones deben tener ambas el mismo número de elementos, parámetros, argumentos o aridad.

Ejemplos:

T1 = f(X, Y, Z) unifica con T2 = f(1,2,3) ya que sus funciones tiene el mismo nombre (f) y cuentan con el mismo número de argumentos.

T1 = suma(X, Y) unifica con T2 = suma(1,2) ya que sus funciones tiene el mismo nombre (f) y cuentan con el mismo número de argumentos.

Otras posibles preguntas:

5. Explicar qué es un λ-término y cuáles son sus propiedades fundamentales.

Los términos de λ-cálculos o **λ-términos** son utilizados para definir funciones dentro del paradigma funcional estando definidos por las siguientes propiedades fundamentales:

- **Confluencia**. Si dos expresiones se pueden reducir (confluyen) a una misma es que son equivalente.

- **Normalización**. Un λ-término es normalizable si es interpretable hasta sus últimas consecuencias, es decir, un λ-término normalizable es aquel que ha sido "simplificado" por completo.

  Normalizar consiste en una aplicación sucesiva de reglas de reducción sobre un λ-término hasta llegar a un término en forma normal. Diremos que un término tiene forma normal si su normalización termina.

6. Define el Teorema de Church-Rosser.

El Teorema de Church-Rosser expone que una **β-reducción** es confluente si M (elemento del conjunto posible de combinaciones del lenguaje, $T_{\Sigma}(x)$) es normalizable.