PEC 1. 22 de Marzo de 2012

1.- *(*0,75 *ptos.)* Decir si las siguientes afirmaciones son ciertas o falsas razonando adecuadamente la respuesta:

a) La fórmula $\exists X p(X) \rightarrow p(a)$ es válida.

b) Sea S un conjunto de cláusulas del Cálculo Proposicional y supongamos que un literal l aparece en alguna cláusula de S y que l^c no aparece en ninguna cláusula de S. Sea S' la forma clausal obtenida a partir de S eliminando todas las cláusulas en las que aparece l. Entonces, $S \approx S'$.

2.- *(*0,75 *ptos.)* Estudiar la validez de la fórmula

$$((p \rightarrow q) \rightarrow r) \rightarrow (q \rightarrow r)$$

mediante el uso de tableros semánticos.

3.- *(*0,75 *ptos.)* ¿De las fórmulas

$$\forall X \forall Y [p(X) \rightarrow r(X,Y)],$$

$$\exists X \exists Y [\neg(\neg q(Y) \vee r(X,Y)))] \text{ y}$$

$$\forall X \exists Y [p(X) \vee s(X,Y)],$$

se deduce $\exists X \exists Y r(X,Y)$? Usar resolución para responder a la pregunta.

4.- *(*0,75 *ptos.)* Sea P el programa lógico dado por:

$$C_1 = p(X,Y) \leftarrow q(Z,X), r(Z,V), q(V,Y).$$
$$C_2 = p(X,Y) \leftarrow q(Z,Y), r(Z,V), q(V,X).$$

$$C_3 = q(a,c). \qquad C_4 = q(a,d). \qquad C_5 = r(a,b).$$
$$C_6 = r(b,a). \qquad C_7 = q(b,e).$$

Computar las respuestas correctas para la pregunta $\leftarrow p(c,Y)$ usando resolución SLD a la Prolog.

PEC 1. 22 de Marzo de 2012

1.- *(0,75 ptos.)* Decir si las siguientes afirmaciones son ciertas o falsas razonando adecuadamente la respuesta:

a) Sea $A(X_1, \ldots, X_n)$ una fórmula sin cuantificadores y sin símbolos de función. Entonces, la fórmula
$$\forall X_1 \cdots \forall X_n A(X_1, \ldots, X_n)$$
es satisfacible si y solo si es satisfacible para una interpretación cuyo dominio se reduce a un único elemento.

b) Sea $C = \{l\} \in S$ una cláusula unitaria y sea S' la forma clausal obtenida de S eliminando las cláusulas que contienen a l y eliminando l^c de las restantes. Entonces, $S \approx S'$.

2.- *(0,75 ptos.)* Probar la validez de la fórmula

$$(\neg p \wedge q \wedge (\neg p \to (q \to r)) \wedge ((q \wedge r) \to s)) \to s$$

mediante el uso de tableros semánticos.

3.- *(0,75 ptos.)* ¿De las fórmulas

$$\forall X \forall Y (\neg p(X) \vee r(X,Y)),$$

$$\neg(\forall X \forall Y (q(Y) \to r(X,Y)) \text{ y}$$

$$\forall X \exists Y (p(X) \vee s(X,Y))),$$

se deduce $\exists X \exists Y r(X,Y)$? Usar resolución para responder a la pregunta.

4.- *(0,75 ptos.)* Sea P el programa lógico dado por:

$$\begin{aligned} &C_1 = p(X,Z) \leftarrow q(X,Y), p(Y,Z). \\ &C_2 = p(X,X). \\ &C_3 = q(a,b). \end{aligned}$$

Computar las respuestas correctas para la pregunta $\leftarrow p(X,b)$ usando resolución SLD a la Prolog.

PEC 2. 10 de Mayo de 2012

1.- *(0,75 ptos.)* Responder a la siguiente pregunta razonando adecuadamente la respuesta: ¿Es el número de combinadores de punto fijo finito?

2.- *(0,75 ptos.)* Dados los $\lambda-$términos

$$\begin{array}{lcl} \textbf{true} & \equiv & \lambda xy.x \\ \textbf{false} & \equiv & \lambda xy.y \\ \textbf{if} & \equiv & \lambda pxy.pxy, \end{array}$$

escribir un $\lambda-$término **nor** que codifique el conectivo lógico $\downarrow$. Verificar, con todo detalle, la corrección de la citada codificación.

3.- *(0,75 ptos.)* Considerar el siguiente fragmento W de un programa:

```
si B1 entonces   P1
      si no      si B2  entonces P2
                        si no P3
                 fsi
fsi
```

Se pide:

a) Dada una postcondición cualquiera q, calcular una ecuación que relacione $wp(W, q)$ con $wp(\mathtt{P1}, q)$, $wp(\mathtt{P2}, q)$ y $wp(\mathtt{P3}, q)$.

b) Generalizar la fórmula obtenida en a) para un número arbitrario de expresiones `si` anidadas.

4.- *(0,75 ptos.)* Demostrar que el siguiente programa calcula el máximo común divisor de dos números enteros positivos a y b (corrección parcial):

$\{a > 0 \ \wedge \ b > 0\}$

```
    x:=a; y:= b; z:=1;
      mientras y ≠ 0 hacer
        casos
          par(x) ∧ par(y) → x:=x/2 ; y:=y/2 ; z:=2*z
          par(x) ∧ impar(y) → x:=x/2
          impar(x) ∧ par(y) → y:=y/2
          impar(x) ∧ impar(y) ∧ (x≥y) → t:=(x-y)/2 ; x:=y ; y:= t
          impar(x) ∧ impar(y) ∧ (y>x) → t:=(y-x)/2 ; y:=t
        fcasos
      fmientras
```

$\{z \cdot x = mcd(a, b)\}$

PEC 2. 10 de Mayo de 2012

1.- *(0,75 pto.)* Responder a la siguiente pregunta razonando adecuadamente la respuesta: Supongamos dados dos $\lambda-$términos M y N tales que $M = N$. ¿Es cierto que, entonces, $M \twoheadrightarrow N$ ó $N \twoheadrightarrow M$?

2.- *(0,75 ptos.)* Dados los $\lambda-$términos

$$\begin{array}{lcl} \textbf{true} & \equiv & \lambda xy.x \\ \textbf{false} & \equiv & \lambda xy.y \\ \textbf{if} & \equiv & \lambda pxy.pxy, \end{array}$$

escribir un $\lambda-$término **nand** que codifique el conectivo lógico $|$. Verificar, con todo detalle, la corrección de la citada codificación.

3.- *(0,75 ptos.)* Considerar el siguiente fragmento W de un programa (**Nota**: Se recuerda que en `casos`, las expresiones booleanas `B1`, `B2` y `B3` son disjuntas y cubren todas las posibilidades.):

```
casos
  B1 -> P1
  B2 -> P2
  B3 -> P3
fcasos
```

a) Dada una postcondición cualquiera q, calcular una ecuación que relacione $wp(W, q)$ con $wp(\texttt{P1}, q)$, $wp(\texttt{P2}, q)$ y $wp(\texttt{P3}, q)$.

b) Supongamos ahora que q es un invariante para cada una de las expresiones `Si`, es decir $wp(\texttt{Si}, q) = q$ para $i = 1, 2, 3$. Decidir si, en ese caso, q también es invariante de W, esto es, $wp(W, q) = q$

4.- *(0,75 ptos.)* Demostrar que el siguiente programa calcula el máximo común divisor de dos números enteros positivos a y b (corrección parcial):

$\{a > 0 \ \wedge \ b > 0\}$

```
 x:=a; y:= b; z:=1
   mientras y ≠ 0 hacer
     casos
      par(x) ∧ par(y) → x:=x/2 ; y:=y/2 ; z:= 2*z
      par(x) ∧impar(y) → x:=x/2
      impar(x) ∧ par(y) → y:=y/2
      impar(x) ∧ impar(y) ∧ (x≥y) → t:=(x-y)/2 ; r:=(x+y)/2 ; x:=r; y:= t
      impar(x) ∧ impar(y) ∧ (y>x) → t:=(y-x)/2 ; r:=(x+y)/2 ; x:=r ; y:= t
     fcasos
   fmientras
```

$\{z \cdot x = mcd(a, b)\}$

Examen Extraordinario. 29 de Junio de 2012

1.- *(2 ptos.)* Se pide:

a) Demostrar que si C_1 y C_2 son cláusulas (del cálculo proposicional) generatrices, su resolvente es satisfacible si y solo C_1 y C_2 son simultáneamente satisfacibles.

b) Dar un ejemplo que demuestre que el hecho de que dos $\lambda-$términos sean iguales no implica que uno de los dos se reduzca al otro.

2.- *(1 pto.)* Utilizar tableros semánticos para probar la validez de la fórmula:

$$(\forall X(p(X) \wedge q(X))) \leftrightarrow (\forall Xp(X) \wedge \forall Xq(X))$$

3.- *(1 pto.)* Las listas se pueden codificar, empleando el Cálculo de Predicados, mediante una función . de aridad 2, y una constante *nil*, la lista vacía, de manera que: una lista con un elemento se representa mediante $.(a, nil)$, con dos elementos $.(a, .(b, nil)), \ldots$ y así sucesivamente. Definir un procedimiento lógico *append* de aridad 3 que tome valor cierto si el tercer elemento es la concatenación de los dos primeros.
Llevar a cabo resolución SLD a la Prolog, para obtener las dos primeras respuestas correctas a la pregunta:

$$\leftarrow append(Y, Z, .(a, .(b, .(c, nil))))$$

4.- *(1 pto.)* Probar que el $\lambda-$término Θ definido por:

$$\begin{array}{rcl} A & \equiv & \lambda xy.y(xxy) \\ \Theta & \equiv & AA \end{array}$$

es un combinador de punto fijo. Usando Θ, definir un $\lambda-$término que calcule el resto de la división de dos número naturales positivos codificados a la Church. (**Nota**: No hace falta redefinir todos los $\lambda-$términos involucrados que ya hayan sido definidos en clase.)

5.- *(1 pto.)* Demostrar que el siguiente programa, que toma como entrada un entero a y una lista de enteros L, devuelve una lista que es la lista L una vez eliminadas todas las apariciones de a en la misma (corrección parcial) (**NOTA**: Se asume que se dispone de las funciones correctas `reverse`, `cons`, `hd` y `tl`):

```
{T}
 L1:=reverse(L); L2:= nil;
   mientras (L1 ≠ nil) hacer
      si hd(L1)= a entonces L1:= tl(L1)
         si no L2:=cons(hd(L1),L2); L1:=tl(L1)
      fsi
   fmientras
{L2 = eliminar(a, L)}
```

PEC 1. 21 de Marzo de 2013

1.- *(*0,75 *ptos.)* Decir si las siguientes afirmaciones son ciertas o falsas razonando adecuadamente la respuesta:

a) Sea $A(X_1, \ldots, X_n)$ una fórmula sin cuantificadores y sin símbolos de función. Entonces, la fórmula

$$\forall X_1 \cdots \forall X_n A(X_1, \ldots, X_n)$$

es satisfacible si y solo si es satisfacible para una interpretación cuyo dominio se reduce a un único elemento.

b) Sea $C = \{l\} \in S$ una cláusula unitaria y sea S' la forma clausal obtenida de S eliminando las cláusulas que contienen a l y eliminando l^c de las restantes. Entonces, $S \approx S'$.

2.- *(*0,75 *ptos.)* Construir un tablero semántico para la fórmula

$$((p \oplus q) \rightarrow \neg(\neg q \rightarrow r)) \wedge (p \vee q)$$

Deducir si la fórmula es satisfacible.

3.- *(*0,75 *ptos.)* ¿De las fórmulas

$$\forall X \forall Y p(X) \rightarrow (r(X) \wedge s(X, Y)),$$

$$\forall X \exists Y \forall Z \neg s(X, Y) \vee t(X, Z) \text{ y}$$

$$\forall Y \forall Z \neg t(Y, Z) \vee v(Y),$$

se deduce $\forall X p(X) \rightarrow v(X)$? Usar resolución para responder a la pregunta.

4.- *(*0,75 *ptos.)* Las listas se pueden codificar, empleando el Cálculo de Predicados, mediante una función . de aridad 2, y una constante nil, la lista vacía, de manera que: una lista con un elemento se representa mediante $.(a, nil)$, con dos elementos $.(a, .(b, nil)), \ldots$ y así sucesivamente. Definir un procedimiento lógico $reverse$ de aridad 2 que tome valor cierto si el segundo elemento es la reflexión del primero.

Llevar a cabo resolución SLD a la Prolog, para obtener una respuesta correcta a la pregunta:

$$\leftarrow reverse(.(a, .(b, .(c, nil))), L)$$

PEC 1. 21 de Marzo de 2013

1.- *(0,75 ptos.)* Decir si las siguientes afirmaciones son ciertas o falsas razonando adecuadamente la respuesta:

a) La fórmula $\exists X p(X) \rightarrow p(a)$ es válida.

b) Sea S un conjunto de cláusulas del Cálculo Proposicional y supongamos que un literal l aparece en alguna cláusula de S y que l^c no aparece en ninguna cláusula de S. Sea S' la forma clausal obtenida a partir de S eliminando todas las cláusulas en las que aparece l. Entonces, $S \approx S'$.

2.- *(0,75 ptos.)* Estudiar la validez de la fórmula

$$((p \rightarrow (q \rightarrow r)) \wedge ((r \wedge s) \rightarrow t) \wedge (\neg w \rightarrow (s \wedge \neg t))) \rightarrow (p \rightarrow (q \rightarrow w))$$

mediante el uso de tableros semánticos.

3.- *(0,75 ptos.)* ¿De las fórmulas

$$\forall X \forall Y p(X) \rightarrow (r(X) \wedge s(X,Y)),$$

$$\forall X \exists Y \forall Z \neg s(X,Y) \vee t(Y,Z) \text{ y}$$

$$\forall Y \forall Z \neg t(Y,Z) \vee v(Y),$$

se deduce $\forall X p(X) \rightarrow v(X)$?
Usar resolución para responder a la pregunta.

4.- *(0,75 ptos.)* Sea P el programa lógico dado por:

$$\begin{array}{l} C_1 : q(X,Y) \leftarrow p(Z,X), p(Z,Y) \\ C_2 : s(X,Y) \leftarrow r(X,Z), q(Z,Y) \\ C_3 : s(X,Y) \leftarrow r(Y,Z), q(Z,X) \end{array}$$

$$\begin{array}{lll} C_4 = p(a,b). & C_5 = p(a,c). & C_6 = r(b,d). \\ C_7 = r(d,b). & C_8 = p(d,e). & \end{array}$$

Computar las respuestas correctas para la pregunta $\leftarrow s(c,X)$ usando resolución SLD a la Prolog.

Examen extraordinario. 12 de Junio de 2013

1.- Se pide:

a) (0,6 *ptos.)* Demostrar que la fórmula $A_1 \wedge A_2 \wedge A_3$, donde

$$A_1 = \forall X \exists Y p(X,Y)$$

$$A_2 = \forall X \neg p(X,X)$$

$$A_3 = \forall X \forall Y \forall Z (p(X,Y) \wedge p(Y,Z)) \to p(X,Z)$$

no admite un modelo finito.

b) (0,7 *ptos.)* Demostrar que el número de combinadores de punto fijo no es finito.

c) (0,7 *ptos.)* Demostrar que si dos $\lambda-$términos, M y N, son iguales, no necesariamente $M \twoheadrightarrow N$ o $N \twoheadrightarrow M$.

2.- (1,5 *ptos.)* Probar la validez de la fórmula

$$(\neg p \wedge q \wedge (\neg p \to (q \to r)) \wedge ((q \wedge r) \to s)) \to s$$

mediante el uso de tableros semánticos.

3.- (1,5 *ptos.)* Las listas se pueden codificar, empleando el Cálculo de Predicados, mediante una función . de aridad 2, y una constante nil, la lista vacía, de manera que: una lista con un elemento se representa mediante $.(a, nil)$, con dos elementos $.(a, .(b, nil)), \ldots$ y así sucesivamente. Definir un procedimiento lógico $reverse$ de aridad 2 que tome valor cierto si el segundo elemento es la reflexión del primero.
Llevar a cabo resolución SLD a la Prolog, para obtener una respuesta correcta a la pregunta:

$$\leftarrow reverse(.(a, .(b, .(c, nil))), L)$$

4.- (1 *pto.)* Dados los $\lambda-$términos

$$\begin{array}{lcl} \textbf{true} & \equiv & \lambda xy.x \\ \textbf{false} & \equiv & \lambda xy.y \\ \textbf{if} & \equiv & \lambda pxy.pxy, \end{array}$$

escribir un $\lambda-$término **implica** que codifique el conectivo lógico $\longrightarrow$. Verificar, con todo detalle, la corrección de la citada codificación.

PEC 1. 13 de Marzo de 2014

1.- (0,75 *ptos.)* Responder, razonadamente, a las siguientes preguntas:

a) ¿Por qué se exige en la regla de resolución del Cálculo de Predicados que las cláusulas generatrices no tengan variables en común?

b) Sea $C = \{l\} \in S$ una cláusula unitaria y sea S' la forma clausal obtenida de S eliminando las cláusulas que contienen a l y eliminando l^c de las restantes. ¿Es cierto que $S \approx S'$?

2.- (0,75 *ptos.)* Estudiar la validez de la fórmula

$$((p \to (r \vee q)) \wedge (\neg p \to \neg s) \wedge s) \to r$$

mediante el uso de tableros semánticos.

3.- (0,75 *ptos.)* ¿De las fórmulas

$$\forall X \forall Y (p(X,Y) \to r(Y)),$$

$$\forall X \exists Y (p(X,Y) \vee \neg q(X,Y)) \text{ y}$$

$$\exists X \forall Y (q(X,Y) \wedge s(Y,X)),$$

se deduce $\exists X \neg r(X)$? Usar resolución para responder a la pregunta.

4.- (0,75 *ptos.)* Las listas se pueden codificar, empleando el Cálculo de Predicados, mediante una función . de aridad 2, y una constante nil, la lista vacía, de manera que: una lista con un elemento se representa mediante $.(a, nil)$, con dos elementos $.(a, .(b, nil)), \ldots$ y así sucesivamente. Escribir un programa lógico que defina un predicado $interseccion$ de aridad 3 que tome valor cierto si el tercer elemento es la intersección, sin repeticiones, de los dos primeros.

Llevar a cabo resolución SLD a la Prolog para obtener **una** respuesta correcta a la pregunta:

$$\leftarrow interseccion(.(1, nil), .(1, .(3, nil)), Z).$$

NOTA: En el programa lógico, si es necesario, puede utilizarse el siguiente procedimiento

$$pertenece(X, .(X, L)).$$
$$pertenece(X, .(Y, L)) \leftarrow pertenece(X, L).$$

PEC 1. 13 de Marzo de 2014

1.- *(0,75 ptos.)* Responder, razonadamente, si las siguientes afirmaciones son correctas:

a) Supongamos dada $A(X_1, \ldots, X_n)$ una fórmula sin cuantificadores y sin símbolos de función y sea F la fórmula

$$\forall X_1 \cdots \forall X_n A(X_1, \ldots, X_n).$$

Entonces, F es satisfacible si y solo si es satisfacible para una interpretación cuyo dominio se reduce a un único elemento.

b) Si dos cláusulas generatrices del Cálculo Proposicional son simultáneamente satisfacibles, entonces su resolvente también lo es.

2.- *(0,75 ptos.)* Estudiar la validez de la fórmula

$$((p \to q) \land ((r \land t) \to p) \land q \land r) \to \neg t$$

mediante el uso de tableros semánticos.

3.- *(0,75 ptos.)* ¿De las fórmulas

$$\forall X \forall Y(\neg p(X, Y) \lor r(Y)),$$

$$\forall X \exists Y(q(X, Y) \to p(X, Y)) \text{ y}$$

$$\exists X \forall Y(q(X, Y) \land s(Y, X)),$$

se deduce $\neg(\forall X \neg r(X))$? Usar resolución para responder a la pregunta.

4.- *(0,75 ptos.)* Las listas se pueden codificar, empleando el Cálculo de Predicados, mediante una función . de aridad 2, y una constante nil, la lista vacía, de manera que: una lista con un elemento se representa mediante $.(a, nil)$, con dos elementos $.(a, .(b, nil)), \ldots$ y así sucesivamente. Escribir un programa lógico que defina un predicado *sin-repeticiones* de aridad 2 que tome valor cierto si el segundo elemento es una lista que contiene los mismos elementos que la primera, pero sin repetir ninguno.

Llevar a cabo resolución SLD a la Prolog para obtener **una** respuesta correcta a la pregunta:

$$\leftarrow \textit{sin-repeticiones}(.(a, .(a, nil)), Z).$$

NOTA: En el programa lógico, si es necesario, puede utilizarse el siguiente procedimiento

$pertenece(X, .(X, L)).$
$pertenece(X, .(Y, L)) \leftarrow pertenece(X, L).$

PEC 2. 8 de Mayo de 2014

1.- *(0,75 ptos.)* Responder a las siguientes preguntas razonando adecuadamente la respuesta:

i) ¿Por qué se exige para que la sutitución $M[N/x]$ sea correcta que

$$BV(M) \bigcap FV(N) = \emptyset?$$

ii) ¿Todo $\lambda-$término admite forma normal?

2.- *(0,75 ptos.)* Probar que el $\lambda-$término Θ definido por:

$$\begin{array}{rcl} A & \equiv & \lambda xy.y(xxy) \\ \Theta & \equiv & AA \end{array}$$

es un combinador de punto fijo. Usando Θ, definir un $\lambda-$término que calcule la suma de todos los elementos de una lista de números enteros. (**Nota**: No hace falta redefinir todos los $\lambda-$términos involucrados que ya hayan sido definidos en clase. Por ejemplo, si se necesita, no es necesario redifinir el $\lambda-$término **null** que determina si una lista es vacía. Por otro lado, por convenio se define la suma de los elementos de una lista vacía como cero.)

3.- *(0,75 ptos.)* Considerar el siguiente fragmento W de un programa:

```
si B1 entonces   P1
      si no      si B2  entonces P2
                        si no P3
                 fsi
fsi
```

Se pide:

a) Dada una postcondición cualquiera q, calcular una ecuación que relacione $wp(W, q)$ con $wp(\texttt{P1}, q)$, $wp(\texttt{P2}, q)$ y $wp(\texttt{P3}, q)$.

b) Generalizar la fórmula obtenida en a) para un número arbitrario de expresiones `si` anidadas.

4.- *(0,75 ptos.)* Demostrar que el siguiente programa, que toma como entrada dos enteros m y n, con $m \geq 0$ y $n > 0$, calcula el cociente de la división euclídea de m por n (corrección parcial):

$\{m \geq 0, n > 0\}$

```
    x:=m; z:= 0;
      mientras  (x ≥ n) hacer
         z:= z + 1; x:= x - n
      fmientras
```

$\{z \; = \; coc(m, n)\}$

PEC 2. 8 de Mayo de 2014

1.- (0,75 *ptos.)* Responder a las siguientes preguntas razonando adecuadamente la respuesta:

i) Enunciar el Teorema de Church-Rosser. ¿Qué consecuencias se pueden extraer del mismo con respecto a la igualdad de $\lambda-$términos y formas normales?

ii) ¿Todo $\lambda-$término admite forma normal?

2.- (0,75 *ptos.)* Probar que el $\lambda-$término Θ definido por:

$$\begin{array}{rcl} A & \equiv & \lambda xy.y(xxy) \\ \Theta & \equiv & AA \end{array}$$

es un combinador de punto fijo. Usando Θ, definir un $\lambda-$término que calcule el cociente de la división de dos número naturales positivos codificados a la Church. (**Nota**: No hace falta redefinir todos los $\lambda-$términos involucrados que ya hayan sido definidos en clase. Por ejemplo, si se necesita, no hace falta redefinir el $\lambda-$término **iszero** que determina si un natural codificado a la Church es nulo.)

3.- (0,75 *ptos.)* Considerar el siguiente fragmento W de un programa (**Nota**: Se recuerda que en `casos`, las expresiones booleanas `B1`, `B2` y `B3` son disjuntas y cubren todas las posibilidades.):

```
casos
  B1 -> P1
  B2 -> P2
  B3 -> P3
fcasos
```

a) Dada una postcondición cualquiera q, calcular una ecuación que relacione $wp(W, q)$ con $wp(\texttt{P1}, q)$, $wp(\texttt{P2}, q)$ y $wp(\texttt{P3}, q)$.

b) Supongamos ahora que q es un invariante para cada una de las expresiones `Pi`, es decir $wp(\texttt{Pi}, q) = q$ para $i = 1, 2, 3$. Decidir si, en ese caso, q también es invariante de W, esto es, $wp(W, q) = q$

4.- (0,75 *ptos.)* Demostrar que el siguiente programa, que toma como entrada cualquier lista a formada por números enteros, calcula la suma de los mismos (corrección parcial):

$\{T\}$
```
m:=0; L:= a;
  mientras (L ≠ nil) hacer
    m:= m + hd(L); L:= tl(L)
  fmientras
```
$\{m = suma(a)\}$

NOTA: Por convenio, se asume que $suma(nil) = 0$.

Examen Final. 14 de Mayo de 2014

1.- *(2 ptos.)* Decir si las siguientes afirmaciones son ciertas o falsas razonando adecuadamente la respuesta:

a) Sea $A(X_1, \ldots, X_n)$ una fórmula sin cuantificadores y sin símbolos de función. Entonces, la fórmula

$$\forall X_1 \cdots \forall X_n A(X_1, \ldots, X_n)$$

es satisfacible si y solo si es satisfacible para una interpretación cuyo dominio se reduce a un único elemento.

b) El número de combinadores de punto fijo es finito.

2.- *(1 pto.)* Utilizar tableros semánticos para probar la validez de la fórmula:

$$(\forall X(p(X) \wedge q(X))) \leftrightarrow (\forall X p(X) \wedge \forall X q(X))$$

3.- *(1 pto.)* Las listas se pueden codificar, empleando el Cálculo de Predicados, mediante una función . de aridad 2, y una constante *nil*, la lista vacía, de manera que: una lista con un elemento se representa mediante $.(a, nil)$, con dos elementos $.(a, .(b, nil)), \ldots$ y así sucesivamente. Definir un procedimiento lógico *reverse* de aridad 2 que tome valor cierto si el segundo elemento es la reflexión del primero.
Llevar a cabo resolución SLD a la Prolog, para obtener una respuesta correcta a la pregunta:

$$\leftarrow reverse(.(a, .(b, .(c, nil))), L)$$

4.- *(1 pto.)* Probar que el $\lambda-$término $\mathbf{Y}$ definido por:

$$\mathbf{Y} \equiv \lambda f.(\lambda x.f(xx))(\lambda x.f(xx)).$$

es un combinador de punto fijo. Usando $\mathbf{Y}$, definir un $\lambda-$término que calcule el cociente de la división de dos número naturales positivos codificados a la Church. (**Nota**: No hace falta redefinir todos los $\lambda-$términos involucrados que ya hayan sido definidos en clase.)

5.- *(1 pto.)* Demostrar que el siguiente programa, que toma como entrada un entero a y una lista de enteros L, devuelve una lista que es la lista L una vez eliminadas todas las apariciones de a en la misma (corrección parcial) (**NOTA**: Se asume que se dispone de las funciones correctas `reverse`, `cons`, `hd` y `tl`):

```
{T}
 L1:=reverse(L); L2:= nil;
   mientras (L1 ≠ nil) hacer
     si hd(L1)= a entonces L1:= tl(L1)
       si no L2:=cons(hd(L1),L2); L1:=tl(L1)
     fsi
   fmientras
{L2 = eliminar(a, L)}
```

Examen Extraordinario. 18 de Junio de 2014

1.- *(2 ptos.)* Responder a las siguientes preguntas razonando adecuadamente la respuesta:

i) Enunciar el Teorema de Church-Rosser. ¿Qué consecuencias se pueden extraer del mismo con respecto a la igualdad de $\lambda-$términos y formas normales?

b) ¿Es la fórmula $\exists X p(X) \to p(a)$ válida?

2.- *(1 pto.)* ¿De las fórmulas

$$\forall X \forall Y [p(X) \to r(X,Y)],$$

$$\exists X \exists Y [\neg(\neg q(Y) \lor r(X,Y)))] \text{ y}$$

$$\forall X \exists Y [p(X) \lor s(X,Y)],$$

se deduce $\exists X \exists Y r(X,Y)$? Usar resolución para responder a la pregunta.

3.- *(1 pto.)* Las listas se pueden codificar, empleando el Cálculo de Predicados, mediante una función . de aridad 2, y una constante nil, la lista vacía, de manera que: una lista con un elemento se representa mediante $.(a, nil)$, con dos elementos $.(a, .(b, nil)), \ldots$ y así sucesivamente. Definir un procedimiento lógico *append* de aridad 3 que tome valor cierto si el tercer elemento es la concatenación de los dos primeros.
Llevar a cabo resolución SLD a la Prolog, para obtener las dos primeras respuestas correctas a la pregunta:

$$\leftarrow append(Y, Z, .(a, .(b, .(c, nil))))$$

4.- *(1 pto.)* Probar que el $\lambda-$término Θ definido por:

$$\begin{array}{rcl} A & \equiv & \lambda xy.y(xxy) \\ \Theta & \equiv & AA \end{array}$$

es un combinador de punto fijo. Usando Θ, definir un
$\lambda-$término **fact** que calcule el factorial de un número natural codificado a la Church. Usar el $\lambda-$término definido para calcular el factorial de dos. (**Nota**: No hace falta redefinir todos los $\lambda-$ términos referentes a los numerales de Church definidos en clase.)

5.- *(1 pto.)* Demostrar que el siguiente programa, que toma como entrada dos listas l_1 y l_2, devuelve una lista que contiene la intersección, sin repeticiones, de las listas l_1 y l_2 (**NOTA**: Se asume que se dispone de las funciones correctas `member` (que verifica si un elemento está o no en una lista), `cons`, `hd` y `tl`):

$\{T\}$

```
 L1:=l1; L2:=l2; L3:= nil;
   mientras (L1 ≠ nil) hacer
      si member(hd(L1),L3) entonces L1:= tl(L1)
         si no si member(hd(L1),L2) entonces L3:=cons(hd(L1),L3); L1:=tl(L1)
            si no L1:=tl(L1)
         fsi
      fsi
   fmientras
```

$\{reverse(L3) = interseccion(l_1, l_2)\}$

PEC 1. 9 de Abril de 2015

1.- *(*0,75 *ptos.)* Decir si las siguientes afirmaciones son ciertas o falsas razonando adecuadamente la respuesta:

a) Toda fórmula satisfacible del Cálculo de Predicados admite un modelo finito.

b) La resolvente C de dos cláusulas generatrices del Cálculo Proposicional es satisfacible si, y sólo si, ambas cláusulas son simultáneamente satisfacibles.

2.- *(*0,75 *ptos.)* Estudiar la validez de la fórmula

$$((p \vee t) \wedge (s \longleftrightarrow t) \wedge (s \longleftrightarrow \neg w)) \longrightarrow (w \longrightarrow p)$$

mediante el uso de tableros semánticos.

3.- *(*0,75 *ptos.)* ¿De las fórmulas

$$\forall X \forall Y (p(X,Y) \rightarrow (r(Y) \vee q(X))),$$

$$\forall X \exists Y (\neg p(X,Y) \longrightarrow \neg s(X,Y)) \text{ y}$$

$$\exists X \forall Y s(Y,X),$$

se deduce $\exists Y r(Y)$? Usar resolución para responder a la pregunta.

4.- *(*0,75 *ptos.)* Las listas se pueden codificar, empleando el Cálculo de Predicados, mediante una función . de aridad 2, y una constante nil, la lista vacía, de manera que: una lista con un elemento se representa mediante $.(a, nil)$, con dos elementos $.(a, .(b, nil)), \ldots$ y así sucesivamente. Escribir un programa lógico que defina un predicado $eliminar(X, L, L_1)$ de aridad 3 que tome valor cierto si L_1 es la reflexión de la lista que resulta de eliminar X de la lista L.

Llevar a cabo resolución SLD a la Prolog para obtener **una** respuesta correcta a la pregunta:

$$\leftarrow eliminar(1, .(1, .(2, .(1, nil))), Z).$$

PEC 2. 21 de Mayo de 2015

1.- *(0,75 ptos.)* Responder a las siguientes preguntas razonando adecuadamente la respuesta:

i) ¿Es finito el número de combinadores de punto fijo?

ii) ¿Por qué se exige para que la sutitución $M[N/x]$ sea correcta que

$$BV(M)\bigcap FV(N)=\emptyset?$$

2.- *(0,75 ptos.)* En $\lambda-$cálculo, y a partir de la codificación de los pares ordenados, se pueden codificar las listas como sigue (se recuerda que **pair** construye pares ordenados y que los términos **fst** y **snd** aplicados a un par ordenado calculan, respectivamente, primera y segunda coordenada):

$$\begin{array}{lcl}
\mathbf{nil} & \equiv & \lambda z.z \\
\mathbf{cons} & \equiv & \lambda xy.\ \mathbf{pair}\ \mathbf{false}(\ \mathbf{pair}\ xy) \\
\mathbf{null} & \equiv & \mathbf{fst} \\
\mathbf{hd} & \equiv & \lambda z.\ \mathbf{fst}\ (\ \mathbf{snd}\ z) \\
\mathbf{tl} & \equiv & \lambda z.\ \mathbf{snd}\ (\ \mathbf{snd}\ z).
\end{array}$$

Teniendo en cuenta lo anterior, se pide:

a) Probar que el $\lambda-$término $\mathbf{Y}\equiv\lambda f.(\lambda x.f(xx))(\lambda x.f(xx))$ es un combinador de punto fijo.

b) Definir, usando $\mathbf{Y}$, un $\lambda-$término **invertir** que, aplicado a una lista codificada en $\lambda-$cálculo, calcule su inversa.

3.- *(0,75 ptos.)* Considerar el siguiente fragmento W de un programa:

```
si B1 entonces   P1
      si no      si B2  entonces P2
                        si no P3
                 fsi
fsi
```

Se pide:

a) Dada una postcondición cualquiera q, calcular una ecuación que relacione $wp(W,q)$ con $wp(\mathtt{P1},q)$, $wp(\mathtt{P2},q)$ y $wp(\mathtt{P3},q)$.

b) Generalizar la fórmula obtenida en a) para un número arbitrario de expresiones `si` anidadas.

4.- *(0,75 ptos.)* Probar la corrección (parcial) del siguiente programa (a y b son números enteros y `coc` denota el cociente de la división entera):

$$\{a > 0;\ b \geq 0\}$$

$$\texttt{S:=} \begin{pmatrix} a & b \\ 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$\texttt{mientras } \mathtt{s}_{12} \neq 0 \texttt{ hacer}$$

$$\texttt{q:=coc(}\mathtt{s}_{11}\texttt{,}\mathtt{s}_{12}\texttt{); Q:=}\begin{pmatrix} 0 & 1 \\ 1 & -\mathtt{q} \end{pmatrix}\texttt{; S:=S*Q}$$

$$\texttt{fmientras}$$

$$\{s_{21}a + s_{31}b = s_{11} \ \wedge\ s_{11} = mcd(a,b)\}$$

Examen Final. 21 de Mayo de 2015

1.- *(2 ptos.)* Decir si las siguientes afirmaciones son ciertas o falsas razonando adecuadamente la respuesta:

a) La resolvente de dos cláusulas generatrices en el Cálculo Proposicional es satisfacible si, y sólo si, las cláusulas son simultáneamente satisfacibles.

b) El número de combinadores de punto fijo es finito.

2.- *(1 pto.)* ¿De las fórmulas

$$\forall X \forall Y (p(X,Y) \rightarrow (r(Y) \vee q(X))),$$

$$\forall X \exists Y (\neg p(X,Y) \longrightarrow \neg s(X,Y)) \text{ y}$$

$$\exists X \forall Y s(Y,X),$$

se deduce $\exists Y r(Y)$? Usar resolución para responder a la pregunta.

3.- *(1 pto.)* Las listas se pueden codificar, empleando el Cálculo de Predicados, mediante una función . de aridad 2, y una constante nil, la lista vacía, de manera que: una lista con un elemento se representa mediante $.(a, nil)$, con dos elementos $.(a, .(b, nil)), \ldots$ y así sucesivamente. Escribir un programa lógico que defina un predicado $union(L, L_1, L_2)$ de aridad 3 que tome valor cierto si L_2 es la unión de las listas L y L_1. Por ejemplo, $union(.(1, nil), .(1, .(2, nil)), .(1, .(2, nil)))$ o $union(.(1, nil), .(2, .(2, nil)), .(1, .(2, .(2, nil))))$ se aceptarían como ciertas.

Llevar a cabo resolución SLD a la Prolog para obtener **una** respuesta correcta a la pregunta:

$$\leftarrow union(.(1, .(2, nil)), .(2, .(3, nil)), Z).$$

4.- *(1 pto.)* En $\lambda-$cálculo, y a partir de la codificación de los pares ordenados, se pueden codificar las listas como sigue (se recuerda que **pair** construye pares ordenados y que los términos **fst** y **snd** aplicados a un par ordenado calculan, respectivamente, primera y segunda coordenada):

$$\begin{array}{lcl}
\mathbf{nil} & \equiv & \lambda z.z \\
\mathbf{cons} & \equiv & \lambda xy.\ \mathbf{pair}\ \mathbf{false}(\ \mathbf{pair}\ xy) \\
\mathbf{null} & \equiv & \mathbf{fst} \\
\mathbf{hd} & \equiv & \lambda z.\ \mathbf{fst}\ (\ \mathbf{snd}\ z) \\
\mathbf{tl} & \equiv & \lambda z.\ \mathbf{snd}\ (\ \mathbf{snd}\ z).
\end{array}$$

Teniendo en cuenta lo anterior, se pide:

a) Probar que el $\lambda-$término $\mathbf{Y} \equiv \lambda f.(\lambda x.f(xx))(\lambda x.f(xx))$ es un combinador de punto fijo.

b) Definir, usando $\mathbf{Y}$, un $\lambda-$término **invertir** que, aplicado a una lista codificada en $\lambda-$cálculo, calcule su inversa.

5.- *(1 pto.)* Probar la corrección (parcial) del siguiente programa (a y b son números enteros y `coc` denota el cociente de la división entera):

$\{a > 0;\ b \geq 0\}$

$$\texttt{S:=} \begin{pmatrix} a & b \\ 1 & 0 \\ 0 & 1 \end{pmatrix}$$

`mientras` $\texttt{s}_{12} \neq 0$ `hacer`

$\texttt{q:=coc(s}_{11}\texttt{,s}_{12}\texttt{); Q:=}\begin{pmatrix} 0 & 1 \\ 1 & -\texttt{q} \end{pmatrix}\texttt{; S:=S*Q}$

`fmientras`

$\{s_{21}a + s_{31}b = s_{11}\ \wedge\ s_{11} = mcd(a, b)\}$

Examen Extraordinario. 26 de Junio de 2015

1.- *(2 ptos.)* Responder a las siguientes preguntas razonando adecuadamente la respuesta:

i) ¿Es la fórmula $\exists X p(X) \to p(a)$ válida?

b) Enunciar el Teorema de Church-Rosser. ¿Qué consecuencias se pueden extraer del mismo con respecto a la igualdad de $\lambda-$términos y formas normales?

2.- *(1 pto.)* ¿De las fórmulas

$$\forall X \forall Y (p(X) \to (q(X) \vee r(X,Y))),$$

$$\exists X \forall Y (r(X,Y) \longrightarrow q(Y)) \text{ y}$$

$$\forall Y \exists X (\neg q(X) \vee r(X,Y)),$$

se deduce $\exists X \forall Y (\neg p(X) \vee q(X) \vee r(f(Y),Y))$? Usar resolución para responder a la pregunta.

3.- *(1 pto.)* Las listas se pueden codificar, empleando el Cálculo de Predicados, mediante una función . de aridad 2, y una constante nil, la lista vacía, de manera que: una lista con un elemento se representa mediante $.(a, nil)$, con dos elementos $.(a, .(b, nil)), \ldots$ y así sucesivamente. Escribir un programa lógico que defina un predicado $union(L, L_1, L_2)$ de aridad 3 que tome valor cierto si L_2 es la unión de las listas L y L_1. Por ejemplo, $union(.(1, nil), .(1, .(2, nil)), .(1, .(2, nil)))$ se aceptaría como cierta del mismo modo que $union(.(1, nil), .(2, .(2, nil)), .(1, .(2, .(2, nil))))$.

Llevar a cabo resolución SLD a la Prolog para obtener **una** respuesta correcta a la pregunta:

$$\leftarrow union(.(1, .(2, nil)), .(2, .(3, nil)), Z).$$

4.- *(1 pto.)* Probar que el $\lambda-$término Θ definido por:

$$\begin{array}{lcl} A & \equiv & \lambda xy.y(xxy) \\ \Theta & \equiv & AA \end{array}$$

es un combinador de punto fijo. Usando Θ, definir un $\lambda-$término que calcule el máximo común divisor de dos número naturales codificados a la Church. (**Nota**: No hace falta redefinir todos los $\lambda-$términos involucrados que ya hayan sido definidos en clase.)

5.- *(1 pto.)* Probar la corrección (parcial) del siguiente programa (a y b son números enteros y `coc` denota el cociente de la división entera):

$$\{a > 0;\ b \geq 0\}$$

$$\texttt{S:=}\begin{pmatrix} a & b \\ 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$\texttt{mientras } \texttt{s}_{12} \neq 0 \texttt{ hacer}$$

$$\texttt{q:=coc(s}_{11}\texttt{,s}_{12}\texttt{); Q:=}\begin{pmatrix} 0 & 1 \\ 1 & -\texttt{q} \end{pmatrix}\texttt{; S:=S*Q}$$

$$\texttt{fmientras}$$

$$\{s_{21}a + s_{31}b = s_{11} \ \wedge \ s_{11} = mcd(a,b)\}$$

PEC 1. 31 de Marzo de 2016

1.- *(0,75 ptos.)* Decir si las siguientes afirmaciones son ciertas o falsas razonando adecuadamente la respuesta:

a) Sea $A(X_1, \ldots, X_n)$ una fórmula sin cuantificadores y sin símbolos de función. Entonces, la fórmula

$$\forall X_1 \cdots \forall X_n A(X_1, \ldots, X_n)$$

es satisfacible si, y sólo si, es satisfacible para una interpretación cuyo dominio se reduce a un único elemento.

b) Sea S una fórmula del Cálculo Proposicional insatisfacible. Entonces, todo árbol semántico para la misma ha de tener, al menos, un nodo de inferencia.

2.- *(0,75 ptos.)* Estudiar la validez de la fórmula

$$((p \to (q \to r)) \wedge ((r \wedge s) \to t) \wedge (\neg w \to (s \wedge \neg t))) \to (p \to (q \to w))$$

mediante el uso de tableros semánticos.

3.- *(0,75 ptos.)* ¿De las fórmulas

$$\forall X \forall Y (p(X,Y) \to r(X)),$$

$$\forall X \exists Y (\neg p(X,Y) \longrightarrow \neg s(X,Y)) \text{ y}$$

$$\exists X \forall Y s(X,Y),$$

se deduce $\forall X \exists Y p(X,Y)$? Usar resolución para responder a la pregunta.

4.- *(0,75 ptos.)* Las listas se pueden codificar, empleando el Cálculo de Predicados, mediante una función . de aridad 2, y una constante nil, la lista vacía, de manera que: una lista con un elemento se representa mediante $.(a, nil)$, con dos elementos $.(a, .(b, nil)), \ldots$ y así sucesivamente. Escribir un programa lógico que defina un predicado de aridad cuatro, $union(L, L_1, L_2, L_3)$, que tome valor cierto si L_3 es una lista que tiene por elementos los de la unión conjuntista de las listas, que suponemos no contienen elementos repetidos, L, L_1 y L_2.

Llevar a cabo resolución SLD a la Prolog para obtener **una** respuesta correcta a la pregunta:

$$\leftarrow union(.(1, nil), .(1, .(2, nil)), .(4, nil), Z).$$

PEC 2. 12 de Mayo de 2016

1.- *(0,75 ptos.)* Responder a las siguientes preguntas razonando adecuadamente la respuesta:

i) ¿Todo $\lambda-$término admite forma normal?

ii) ¿Es finito el número de combinadores de punto fijo?

2.- *(0,75 ptos.)* Dados los $\lambda-$términos

$$\begin{array}{lcl} \textbf{true} & \equiv & \lambda xy.x \\ \textbf{false} & \equiv & \lambda xy.y \\ \textbf{if} & \equiv & \lambda pxy.pxy, \end{array}$$

escribir un $\lambda-$término **implica** que codifique el conectivo lógico $\longrightarrow$. Verificar, con todo detalle, la corrección de la citada codificación.

3.- *(0,75 ptos.)* Considerar el siguiente fragmento W de un programa:

```
si B1 entonces   P1
      si no      si B2  entonces P2
                        si no P3
                 fsi
fsi
```

Se pide:

a) Dada una postcondición cualquiera q, calcular una ecuación que relacione $wp(W,q)$ con $wp(\texttt{P1},q)$, $wp(\texttt{P2},q)$ y $wp(\texttt{P3},q)$.

b) Generalizar la fórmula obtenida en a) para un número arbitrario de expresiones `si` anidadas.

4.- *(0,75 ptos.)* Demostrar que el siguiente programa calcula el máximo común divisor de dos números enteros positivos a y b (corrección parcial):

$\{a > 0 \text{ y } b > 0\}$

```
x:=a; y:= b; z:=1;
  mientras y ≠ 0 hacer
    casos
      par(x) y par(y) → x:=x/2 ; y:=y/2 ; z:=2*z
      par(x) y impar(y) → x:=x/2
      impar(x) y par(y) → y:=y/2
      impar(x) y impar(y) y (x≥y) → t:=(x-y)/2 ; x:=y ; y:= t
      impar(x) y impar(y) e (y>x) → t:=(y-x)/2 ; y:=t
    fcasos
  fmientras
```

$\{z \cdot x = mcd(a,b)\}$

Examen Extraordinario. 24 de Junio de 2016

1.- *(2 ptos.)* Responder a las siguientes preguntas razonando adecuadamente la respuesta:

a) ¿Qué relación existe entre los tableros semánticos en el Cálculo Proposicional y la formal normal disyuntiva?

b) Definir la precondición más débil para una expresión `si` y un predicado q. Explicar el porqué de la definición.

2.- *(1 pto.)* ¿De las fórmulas

$$\forall X \forall Y (p(X,Y) \to r(X)),$$

$$\forall X \exists Y (\neg p(X,Y) \longrightarrow \neg s(X,Y)) \text{ y}$$

$$\exists X \forall Y s(X,Y),$$

se deduce $\forall X \exists Y p(X,Y)$? Usar resolución para responder a la pregunta.

3.- *(1 pto.)* Las listas se pueden codificar, empleando el Cálculo de Predicados, mediante una función . de aridad 2, y una constante nil, la lista vacía, de manera que: una lista con un elemento se representa mediante $.(a, nil)$, con dos elementos $.(a, .(b, nil)), \ldots$ y así sucesivamente. Escribir un programa lógico que defina un predicado de aridad cuatro, $cambiar(X, Y, L, L_1)$, que tome valor cierto si L_1 es la lista que resulta de sustituir, en la lista L, todas las apariciones de X por Y.

Llevar a cabo resolución SLD a la Prolog para obtener **una** respuesta correcta a la pregunta:

$$\leftarrow union(1, 2, .(1, .(1, .(3, nil))), Z).$$

4.- *(1 pto.)* En $\lambda-$cálculo, y a partir de la codificación de los pares ordenados, se pueden codificar las listas como sigue (se recuerda que **pair** construye pares ordenados y que los términos **fst** y **snd** aplicados a un par ordenado calculan, respectivamente, primera y segunda coordenada):

$$\begin{array}{lcl}
\textbf{nil} & \equiv & \lambda z.z \\
\textbf{cons} & \equiv & \lambda xy.\ \textbf{pair false}(\ \textbf{pair}\ xy) \\
\textbf{null} & \equiv & \textbf{fst} \\
\textbf{hd} & \equiv & \lambda z.\ \textbf{fst}\ (\ \textbf{snd}\ z) \\
\textbf{tl} & \equiv & \lambda z.\ \textbf{snd}\ (\ \textbf{snd}\ z).
\end{array}$$

Teniendo en cuenta lo anterior, se pide:

a) Probar que el $\lambda-$término $\mathbf{Y} \equiv \lambda f.(\lambda x.f(xx))(\lambda x.f(xx))$ es un combinador de punto fijo.

b) Definir, usando $\mathbf{Y}$, un $\lambda-$término **sumalista** que, aplicado a una lista de enteros codificada en $\lambda-$cálculo, calcule la suma de todos los elementos de la lista. (**Nota**: No hace falta redefinir los $\lambda-$términos ya definidos en clase.)

5.- *(1 pto.)* Probar la corrección (parcial) del siguiente programa (a es un número entero):

$\{a \geq 0\}$
```
x:=0; y:= 1
  mientras y≤a hacer
    x:=x+1; y:= y+2*x+1
  fmientras
```
$\{0 \leq x^2 \leq a < (x+1)^2\}$

PEI 1. 22 de Marzo de 2018

1.- *(*0,75 *ptos.)* Responder a las siguientes preguntas razonando adecuadamente la respuesta:

a) ¿Qué relación existe entre los tableros semánticos en el Cálculo Proposicional y la forma normal disyuntiva?

b) En el proceso de skolenización, ¿cómo se eliminan los cuantificadores existenciales una vez la fórmula en forma normal conjuntiva prenexa? ¿Por qué esa forma de eliminar los cuantificadores existenciales no cambia la satisfabilidad o no satisfabilidad de la fórmula inicial?

2.- *(*0,75 *ptos.)* ¿De las fórmulas

$$\neg(q \vee t)$$

$$(s \longrightarrow r) \vee (s \longrightarrow t)$$

$$r \longrightarrow (p \wedge q)$$

$$\text{y} \quad s \longrightarrow p$$

se deduce $\neg s$? Usar tableros semánticos para responder a esta pregunta.

3.- *(*0,75 *ptos.)* ¿De las fórmulas

$$\forall X \forall Y \forall Z \; p(X,Y) \longrightarrow (q(Y,Z) \longrightarrow r(X,Z)),$$

$$\forall X \exists Y \forall Z \; (r(X,Y) \wedge s(X,Y)) \longrightarrow t(X,Z) \text{ y}$$

$$\exists X \forall Y \; \neg w(X,Y) \longrightarrow (s(X,Y) \wedge \neg t(X,Y)),$$

se deduce $\exists X \forall Y \exists Z \; p(X,Y) \longrightarrow (q(Y,Z) \longrightarrow w(X,Y))$? Usar resolución para responder a la pregunta.

4.- *(*0,75 *ptos.)* Las listas se pueden codificar, empleando el Cálculo de Predicados, mediante una función . de aridad 2, y una constante *nil*, la lista vacía, de manera que: una lista con un elemento se representa mediante $.(a, nil)$, con dos elementos $.(a, .(b, nil)), \ldots$ y así sucesivamente. Escribir un programa lógico que defina un predicado de aridad cuatro, $cambiar(X, Y, L, L_1)$, que tome valor cierto si L_1 es la reflexión de la lista que resulta de sustituir en L todas las apariciones de X por Y. Así, por ejemplo,

$$cambiar(1, 2, .(1, .(3, nil)), .(3, .(2, nil)))$$

se verificaría como cierto.

Llevar a cabo resolución SLD a la Prolog para obtener **una** respuesta correcta a la pregunta:

$$\leftarrow cambiar(a, b, .(a, .(a, .(c, nil))), W).$$

PEI 2. 21 de Mayo de 2018

1.- *(0,75 ptos.)* Responder a las siguientes preguntas razonando adecuadamente la respuesta:

i) ¿Todo $\lambda-$término admite forma normal?

ii) Definir la precondición más débil para una expresión `si` y un predicado q. Explicar el porqué de la definición.

2.- *(0,75 ptos.)* En $\lambda-$cálculo, y a partir de la codificación de los pares ordenados, se pueden codificar las listas como sigue (se recuerda que **pair** construye pares ordenados y que los términos **fst** y **snd** aplicados a un par ordenado calculan, respectivamente, primera y segunda coordenada):

$$\begin{array}{lcl} \textbf{nil} & \equiv & \lambda z.z \\ \textbf{cons} & \equiv & \lambda xy.\ \textbf{pair false}(\ \textbf{pair}\ xy) \\ \textbf{null} & \equiv & \textbf{fst} \\ \textbf{hd} & \equiv & \lambda z.\ \textbf{fst}\ (\ \textbf{snd}\ z) \\ \textbf{tl} & \equiv & \lambda z.\ \textbf{snd}\ (\ \textbf{snd}\ z). \end{array}$$

Teniendo en cuenta lo anterior, se pide:

a) Probar que el $\lambda-$término $\mathbf{Y} \equiv \lambda f.(\lambda x.f(xx))(\lambda x.f(xx))$ es un combinador de punto fijo.

b) Definir, usando $\mathbf{Y}$, un $\lambda-$término **long** que, aplicado a una lista codificada en $\lambda-$cálculo, calcule la longitud de la misma. No es necesario redefinir los $\lambda-$términos utilizados ya definidos en clase.

c) Codificar en $\lambda-$cálculo la lista $L = [1]$. Aplicar el $\lambda-$término **long** a L para calcular su longitud.

3.- *(0,75 ptos.)* Considerar el siguiente fragmento W de un programa:

```
si B1 entonces   P1
      si no      si B2  entonces P2
                        si no  si B3 entonces P3
                                     si no P4
                        fsi
                 fsi
fsi
```

Probar que si I es un invariante para cada uno de los Pi ($i = 1, 2, 3, 4$), entonces es un invariante para W.

4.- Demostrar que el siguiente programa, que toma como entrada dos listas l_1 y l_2 (la segunda sin repeticiones), devuelve una lista cuyos elementos constituyen la unión (conjuntista) de las listas l_1 y l_2 (corrección parcial) (**NOTA**: Se asume que se dispone de las funciones correctas `pertenece` que verifica si un elemento pertenece a una lista, `cons`, `hd` y `tl`):

$\{l_2 \; \textit{sin repeticiones}\}$

```
L1:=l1; L2:= l2 ;
  mientras (L1 ≠ nil) hacer
    si pertenece(hd(L1), L2) entonces L1:= tl(L1)
      si no L2:=cons(hd(L1),L2); L1:=tl(L1)
    fsi
  fmientras
```

$\{L2 = union(l_1, l_2)\}$

Examen Extraordinario. 13 de Junio de 2018

1.- *(2 ptos.)* Responder a las siguientes preguntas razonando adecuadamente la respuesta:

a) ¿Qué relación existe entre los tableros semánticos en el Cálculo Proposicional y la forma normal disyuntiva?

b) ¿Es finito el número de combinadores de punto fijo?

c) Supongamos dados dos $\lambda-$términos, M y N, tales que $M = N$. ¿Es cierto que o bien $M \twoheadrightarrow N$ o $N \twoheadrightarrow M$?

2.- *(1 pto.)* ¿De las fórmulas

$$\forall X \forall Y (p(X,Y) \rightarrow r(X)),$$

$$\forall X \exists Y (\neg p(X,Y) \longrightarrow \neg s(X,Y)) \text{ y}$$

$$\exists X \forall Y s(X,Y),$$

se deduce $\forall X \exists Y p(X,Y)$? Usar resolución para responder a la pregunta.

3.- *(1 pto.)* Las listas se pueden codificar, empleando el Cálculo de Predicados, mediante una función . de aridad 2, y una constante nil, la lista vacía, de manera que: una lista con un elemento se representa mediante $.(a, nil)$, con dos elementos $.(a, .(b, nil)), \ldots$ y así sucesivamente. Escribir un programa lógico que defina un predicado de aridad cuatro, $cambiar(X, Y, L, L_1)$, que tome valor cierto si L_1 es la lista que resulta de sustituir, en la lista L, todas las apariciones de X por Y.

Llevar a cabo resolución SLD a la Prolog para obtener **una** respuesta correcta a la pregunta:

$$\leftarrow cambiar(1, 2, .(1, .(1, .(3, nil))), Z).$$

4.- *(1 pto.)* En $\lambda-$cálculo, y a partir de la codificación de los pares ordenados, se pueden codificar las listas como sigue (se recuerda que **pair** construye pares ordenados y que los términos **fst** y **snd** aplicados a un par ordenado calculan, respectivamente, primera y segunda coordenada):

$$
\begin{array}{lcl}
\mathbf{nil} & \equiv & \lambda z.z \\
\mathbf{cons} & \equiv & \lambda xy.\ \mathbf{pair\ false}(\ \mathbf{pair}\ xy) \\
\mathbf{null} & \equiv & \mathbf{fst} \\
\mathbf{hd} & \equiv & \lambda z.\ \mathbf{fst}\ (\ \mathbf{snd}\ z) \\
\mathbf{tl} & \equiv & \lambda z.\ \mathbf{snd}\ (\ \mathbf{snd}\ z).
\end{array}
$$

Teniendo en cuenta lo anterior, se pide:

a) Probar que el $\lambda-$término $\mathbf{Y} \equiv \lambda f.(\lambda x.f(xx))(\lambda x.f(xx))$ es un combinador de punto fijo.

b) Definir, usando $\mathbf{Y}$, un $\lambda-$término **pertenece** que, aplicado a un número natural codificado a la Church y a una lista de naturales codificada en $\lambda-$cálculo, determine si el número natural pertenece a lista o no. (**Nota**: No hace falta redefinir los $\lambda-$términos ya definidos en clase.)

5.- *(1 pto.)* Probar la corrección (parcial) del siguiente programa (a es un número entero):

$\{a \geq 0\}$

```
   x:=0; y:= 1
     mientras y≤a hacer
        x:=x+1; y:= y+2*x+1
     fmientras
```

$\{0 \leq x^2 \leq a < (x+1)^2\}$

PEI 1. 21 de marzo de 2019

1.- *(0,75 ptos.)* Responder a las siguientes preguntas razonando adecuadamente la respuesta:

a) ¿Por qué se exige en la regla de resolución del Cálculo de Predicados que las cláusulas generatrices no tengan variables en común?

b) ¿Toda fórmula satisfacible en el Cálculo de Predicados admite un modelo finito?

c) ¿Qué relación existe entre tableros semánticos en el Cálculo Proposicional y la forma normal disyuntiva?

2.- *(0,75 ptos.)* ¿De las fórmulas

$$t \longrightarrow q$$

$$p \longrightarrow (r \vee q)$$

$$\neg p \longrightarrow (\neg s \wedge q)$$

$$\text{y} \quad (p \vee r) \longrightarrow t$$

se deduce $s \longrightarrow q$? Usar tableros semánticos para responder a esta pregunta.

3.- *(0,75 ptos.)* ¿De las fórmulas

$$\forall X \forall Y \; p(X,Y) \longrightarrow q(X,Y),$$

$$\forall X \exists Y \; r(X,Y) \longrightarrow s(Y,X) \text{ y}$$

$$\exists X \forall Z \; (\neg r(X,Z) \vee s(X,Z)) \longrightarrow p(X,Z),$$

se deduce $\exists X \exists Y \; p(X,Y) \vee q(X,Y)$? Usar resolución para responder a la pregunta.

4.- *(0,75 ptos.)* Las listas se pueden codificar, empleando el Cálculo de Predicados, mediante una función . de aridad 2, y una constante nil, la lista vacía, de manera que: una lista con un elemento se representa mediante $.(a, nil)$, con dos elementos $.(a, .(b, nil)), \ldots$ y así sucesivamente. Escribir un programa lógico que defina un predicado de aridad dos, $subconjunto(L, L_1)$, que tome valor cierto si, vistas las listas como conjuntos, la lista L es subconjunto de L_1.

Usando el programa definido, levar a cabo resolución SLD a la Prolog para responder a la pregunta

$$\leftarrow subconjunto(.(1, .(1, .(2, nil))), .(1, .(2, .(3, nil)))).$$

Del mismo modo, y usando de nuevo el programa definido anteriormente, realizar resolución SLD a la Prolog para encontrar **una** respuesta correcta a la pregunta

$$\leftarrow subconjunto(.(1, .(2, nil)), L).$$

PEI 2. 17 de mayo de 2019

1.- (0,75 *ptos.)* Responder a las siguientes preguntas razonando adecuadamente la respuesta:

i) Supongamos dados dos $\lambda-$términos M y N tales que $M = N$. ¿Es cierto que, entonces, $M \twoheadrightarrow N$ ó $N \twoheadrightarrow M$?

ii) ¿Por qué se exige para que la sutitución $M[N/x]$ sea correcta que

$$BV(M) \bigcap FV(N) = \emptyset?$$

2.- (0,75 *ptos.)* En $\lambda-$cálculo, y a partir de la codificación de los pares ordenados, se pueden codificar las listas como sigue (se recuerda que **pair** construye pares ordenados y que los términos **fst** y **snd** aplicados a un par ordenado calculan, respectivamente, primera y segunda coordenada):

$$\begin{array}{lcl}
\mathbf{nil} & \equiv & \lambda z.z \\
\mathbf{cons} & \equiv & \lambda xy.\ \mathbf{pair}\ \mathbf{false}(\ \mathbf{pair}\ xy) \\
\mathbf{null} & \equiv & \mathbf{fst} \\
\mathbf{hd} & \equiv & \lambda z.\ \mathbf{fst}\ (\ \mathbf{snd}\ z) \\
\mathbf{tl} & \equiv & \lambda z.\ \mathbf{snd}\ (\ \mathbf{snd}\ z).
\end{array}$$

Teniendo en cuenta lo anterior, se pide:

a) Probar que el $\lambda-$término $\mathbf{Y} \equiv \lambda f.(\lambda x.f(xx))(\lambda x.f(xx))$ es un combinador de punto fijo.

b) Definir, usando $\mathbf{Y}$, un $\lambda-$término **sub** que, aplicado a dos listas de números naturales a la Church codificadas en $\lambda-$cálculo, decida si la primera es subconjunto de la segunda, vistas ambas listas como conjuntos. No es necesario redefinir los $\lambda-$términos utilizados ya definidos en clase.

3.- (0,75 *ptos.)* Considerar el siguiente fragmento W de un programa:

```
si B1 entonces   P1
      si no      si B2  entonces P2
                        si no P3
                 fsi
fsi
```

Probar que si I es un invariante para cada uno de los Pi ($i = 1, 2, 3$), entonces es un invariante para W.

4.- *(0,75 ptos.)* Demostrar que el siguiente programa, que toma como entrada dos listas l_1 y l_2, devuelve una lista que contiene, únicamente, los elementos de la intersección, sin repeticiones, de las listas l_1 y l_2 (**NOTA**: Se asume que se dispone de las funciones correctas `member` (que verifica si un elemento está o no en una lista), `cons`, `hd` y `tl`):

$\{T\}$

```
 L1:=l1; L2:=l2; L3:= nil;
   mientras (L1 ≠ nil) hacer
      si member(hd(L1),L3) entonces L1:= tl(L1)
         si no si member(hd(L1),L2) entonces L3:=cons(hd(L1),L3); L1:=tl(L1)
            si no L1:=tl(L1)
         fsi
      fsi
   fmientras
```

$\{L3 = interseccion(l_1, l_2)\}$

Examen Extraordinario. 19 de junio de 2019

1.- *(2 ptos.)* Decir si las siguientes afirmaciones son correctas, razonando adecuacadamente la respuesta:

a) Sean C_1 y C_2 dos cláusulas generatrices del Cálculo Proposicional y C su resolvente. Entonces, C_1 y C_2 son simultáneamente satisfacibles si, y sólo, si lo es C.

b) Todo $\lambda-$ término admite forma normal.

c) Consideremos el siguiente fragmento, W, de un programa

```
si B1 entonces   P1
     si no       P2
fsi
```

Entonces, si I es un invariante para P1 y P2, también lo es para W.

2.- *(1 pto.)* ¿De las fórmulas

$$\forall X \forall Y (p(X,Y) \rightarrow r(X)),$$

$$\forall X \exists Y (\neg p(X,Y) \longrightarrow \neg s(X,Y)) \text{ y}$$

$$\exists X \forall Y s(X,Y),$$

se deduce $\forall X \exists Y p(X,Y)$? Usar resolución para responder a la pregunta.

3.- *(1 pto.)* Las listas se pueden codificar, empleando el Cálculo de Predicados, mediante una función . de aridad 2, y una constante nil, la lista vacía, de manera que: una lista con un elemento se representa mediante $.(a, nil)$, con dos elementos $.(a, .(b, nil)), \ldots$ y así sucesivamente. Escribir un programa lógico que defina un predicado $eliminar(X, L, L_1)$ de aridad 3 que tome valor cierto si L_1 es la reflexión de la lista que resulta de eliminar X de la lista L.

Llevar a cabo resolución SLD a la Prolog para obtener **una** respuesta correcta a la pregunta:

$$\leftarrow eliminar(1, .(1, .(2, .(1, nil))), Z).$$

4.- *(1 pto.)* En $\lambda-$cálculo, y a partir de la codificación de los pares ordenados, se pueden codificar las listas como sigue (se recuerda que **pair** construye pares ordenados y que los términos **fst** y **snd** aplicados a un par ordenado calculan, respectivamente, primera y segunda coordenada):

$$\begin{array}{lcl}
\textbf{nil} & \equiv & \lambda z.z \\
\textbf{cons} & \equiv & \lambda xy.\ \textbf{pair false}(\ \textbf{pair}\ xy) \\
\textbf{null} & \equiv & \textbf{fst} \\
\textbf{hd} & \equiv & \lambda z.\ \textbf{fst}\ (\ \textbf{snd}\ z) \\
\textbf{tl} & \equiv & \lambda z.\ \textbf{snd}\ (\ \textbf{snd}\ z).
\end{array}$$

Teniendo en cuenta lo anterior, se pide:

a) Probar que el $\lambda-$término $\mathbf{Y} \equiv \lambda f.(\lambda x.f(xx))(\lambda x.f(xx))$ es un combinador de punto fijo.

b) Definir, usando $\mathbf{Y}$, un $\lambda-$término **eliminar** que, aplicado a un número natural, n, codificado a la Church y a una lista de naturales codificada en $\lambda-$cálculo, l, produzca como resultado la lista que resulta de eliminar en l todas las apariciones de n. (**Nota**: No hace falta redefinir los $\lambda-$términos ya definidos en clase.)

c) Codificar en $\lambda-$cálculo la lista $L = [1]$ y aplicar el $\lambda-$término **eliminar** a L y a 1.

5.- *(1 pto.)* Demostrar que el siguiente programa, que toma como entrada dos enteros m y n, con $m \geq 0$ y $n > 0$, calcula cociente y resto de la división euclídea de m por n (corrección parcial):

$\{m \geq 0, n > 0\}$

```
x:=m; z:= 0;
  mientras  (x ≥ n) hacer
    z:= z + 1; x:= x - n
  fmientras
```

$\{z = coc(m,n)\ \wedge\ x = rem(m,n)\}$