Seminario de Matemáticas - Práctica 2

9 de mayo de 2014 Julio Ariel Hurtado Alegría

1. Objetivo General

Complementar el aprendizaje teórico respecto a la lógica proposicional y la lógica de predicados haciendo uso de un lenguaje de programación declarativo cuyo fundamento se encuentra en el razonamiento lógico.

2. Desarrollo

Este es el conjunto de tareas a desarrollar en esta práctica:

1. Desarrollar los ejercicios dirigidos.
2. Analizar y desarrollar los ejercicios propuestos.

3. Ejercicios Dirigidos

(a) Dada una lista de naturales, diseñar e implementar en Prolog un programa que entregue la suma de los elementos de la lista. Ejemplo:

```
?- suma([2,3,4,10],19). => true
?- suma([2,3,4,10],10). => false
```

Solución:
En lógica de primer orden tenemos la siguiente teoría:

a) R1. $(suma(\emptyset) = 0)$

b) R2. $\forall x(suma(\rightarrow x) = x)$

c) R3. $\forall x, l, \exists y, z(y = suma(l), z = y + x \Rightarrow z = suma(x \rightarrow l))$

Solución en Prolog: para solucionarlo entonces nos basamos en el diseño matemático, el cual nos sugiere en R1 que la suma de una lsita vacía es 0. En R2 que la suma de una lista con un elemento es el mismo elemento. R3 nos dice que si tenemos la suma de una lista l, entonces podemos tener la lista aumentada en x cuya suma sería x+z. Esta regla escrita en prolog sería:

```
suma([],0).
suma([X],X).
suma([X|L],Z):-suma(L,Y), Z is Y+X.
```

(b) Defina la función concatenar, que permite concatenar dos listas. Ejemplo:

```
?- concatenar([a,b,c],[d,e,f],X).
X = [a, b, c, d, e, f].
```

Solución:
En lógica de primer orden tenemos la siguiente teoría:

a) R1. $(concatenar(\emptyset, L) = L)$

b) R2. $\forall x, l1, l2 \exists l3 (concatenar(l1, l2) = l3 \Rightarrow concatenar(x \rightarrow l1, l2) = x \rightarrow l3))$

Solución en Prolog: para solucionarlo entonces nos basamos en el diseño matemático, el cual nos sugiere en R1 que la concatenación de una lista con una lista vacía es la misma lista. R2 nos dice que si l3 resulta de concatenar las listas l1 y l2 entonces la concatenación de l1 agregandole un elemento en la cabeza con l2, resultará en l3 agregándole el mismo elemento en la cabeza. Esta regla escrita en prolog sería:

```
concatenar([],L,L).
concatenar([X|L1],L2,[X|L3]):-concatenar(L1,L2,L3).
```

4. Ejercicios Propuestos

- La relación sumaParcial(NumL, Sum) indica en Prolog que NumL es una lista de números y que existe un subconjunto de los números contenidos en ella que sumados dan el número Sum. Ejemplos:

```
?- sumaParcial([7, 2, 8, 5], 15) => true, porque 7+8=15
?- sumaParcial([7, 2, 8, 5], 4) => false
?- sumaParcial([7, 2, 8, 5], 22) => true, porque 7 + 2 + 8 +5= 22
?- sumaParcial([7, 2, 8, 5], 8) => true, porque 8 = 8.
?- sumaParcial([7, 2, 8, 5], 20) => true, porque 7 + 8 + 5 = 20.
?- sumaParcial([7, 2, 8, 5], 0) => true, la suma de elementos en el subconjunto vacío es 0.
```

  Defina las reglas en lógica de primer orden que rigen la relación sumaParcial y luego implementelas en Prolog.

5. Problemas

Diseñe usando lógica de predicados e emplemente en Prolog los siguientes programas:

1. El ordenamiento de una lista siguiendo el enfoque insert-sort
2. Determinar el isomorfismo de dos grafos. Dos grafos $G1(N1, E1)$ y $G2(N2, E2)$ son isomórficos si existe una función biyectiva $f : N1 \rightarrow N2$ tal que para cualquier par de nodos X,Y en N1, X y Y son adyacentes si y solo si f(X) y f(Y) en N2 son adyacentes. Es decir, existe $e_{X,Y} \in E1$ si y solo si, existe $e_{f(X),f(Y)} \in E2$

3. Definir la combinación y cantidad de monedas (billetes) para dar un cambio(devuelto) dado un valor de compra y un valor de pago(con una moneda o billete de mayor denominación al valor pagado).