Universidad Simón Bolívar
Depto. De Computación y Tecnología de la Información
CI3661: Laboratorio de Lenguajes de Programación

Tarea 3

Williams Mariño
Julio De Abreu 05-38072

18 de noviembre de 2012

1. a)

```
%% Implantacion del predicado sumar
sumar(estrella,X,X).
sumar(X,estrella,X).
sumar(up(X),Y,up(Z)):-sumar(X,Y,Z).
```

b)

```
%% Implantacion del predicado restar
restar(estrella,X,X).
restar(up(X), Y, Z):-restar(X,Y,Z).
```

c)

```
%% Implantacion del predicado producto
producto(_,estrella,estrella).
producto(up(estrella),X,X).
producto(X,up(Y),Z):-sumar(X,Y,Z1),producto(X,Z1,Z).
```

Investigación

a) P(x) *and* Q(y)
P(x) : 3
Q(y) : 3
1) P(a) *and* Q(a)
2) P(a) *and* Q(b)
3) P(a) *and* Q(c)
4) P(b) *and* Q(a)
5) P(b) *and* Q(b)
6) P(b) *and* Q(c)
7) P(c) *and* Q(a)
8) P(c) *and* Q(b)
9) P(c) *and* Q(c)

Total: 9 posibles formulas.

b) Tomando las formulas de la parte d), se logro sacar la siguiente cuenta:

18+9+18+18+9+27=99 posibles formulas.

c) P(f(a)) *and* Q(g(a,b))
P(f(a)) *and* Q(g(a,c))

P(f(a)) *and* Q(g(b,c))
P(f(b)) *and* Q(g(a,b))
P(f(b)) *and* Q(g(a,c))
P(f(b)) *and* Q(g(b,c))
P(f(c)) *and* Q(g(a,b))
P(f(c)) *and* Q(g(a,c))
P(f(c)) *and* Q(g(b,c))
P(f(a)) *and* Q(g(b,a))
P(f(a)) *and* Q(g(c,a))
P(f(a)) *and* Q(g(c,b))
P(f(b)) *and* Q(g(b,a))
P(f(b)) *and* Q(g(c,a))
P(f(b)) *and* Q(g(c,b))
P(f(c)) *and* Q(g(b,a))
P(f(c)) *and* Q(g(c,a))
P(f(c)) *and* Q(g(c,b))

P(f(a)) *and* Q(f(a))
P(f(a)) *and* Q(f(b))
P(f(a)) *and* Q(f(c))
P(f(b)) *and* Q(f(a))
P(f(b)) *and* Q(f(b))
P(f(b)) *and* Q(f(c))
P(f(c)) *and* Q(f(a))
P(f(c)) *and* Q(f(b))
P(f(c)) *and* Q(f(c))

P(g(a,b)) *and* Q(g(a,b))
P(g(a,b)) *and* Q(g(a,c))
P(g(a,b)) *and* Q(g(b,c))
P(g(a,b)) *and* Q(g(b,a))
P(g(a,b)) *and* Q(g(c,a))
P(g(a,b)) *and* Q(g(c,b))
P(g(a,c)) *and* Q(g(a,b))
P(g(a,c)) *and* Q(g(a,c))
P(g(a,c)) *and* Q(g(b,c))
P(g(a,c)) *and* Q(g(b,a))
P(g(a,c)) *and* Q(g(b,c))
P(g(a,c)) *and* Q(g(c,a))
P(g(b,c)) *and* Q(g(a,b))
P(g(b,c)) *and* Q(g(a,c))

P(g(b,c)) *and* Q(g(b,c))
P(g(b,c)) *and* Q(g(b,a))
P(g(b,c)) *and* Q(g(b,c))
P(g(b,c)) *and* Q(g(c,a))

P(g(b,a)) *and* Q(g(a,b))
P(g(b,a)) *and* Q(g(a,c))
P(g(b,a)) *and* Q(g(b,c))
P(g(b,a)) *and* Q(g(b,a))
P(g(b,a)) *and* Q(g(c,a))
P(g(b,a)) *and* Q(g(c,b))
P(g(c,a)) *and* Q(g(a,b))
P(g(c,a)) *and* Q(g(a,c))
P(g(c,a)) *and* Q(g(b,c))
P(g(c,a)) *and* Q(g(b,a))
P(g(c,a)) *and* Q(g(b,c))
P(g(c,a)) *and* Q(g(c,a))
P(g(c,b)) *and* Q(g(a,b))
P(g(c,b)) *and* Q(g(a,c))
P(g(c,b)) *and* Q(g(b,c))
P(g(c,b)) *and* Q(g(b,a))
P(g(c,b)) *and* Q(g(b,c))
P(g(c,b)) *and* Q(g(c,a))

Luego, dado que la conjuncion es simetrica, podemos obtener lo siguiente:
Q(f(a)) *and* P(g(a,b))
Q(f(a)) *and* P(g(a,c))
Q(f(a)) *and* P(g(b,c))
Q(f(b)) *and* P(g(a,b))
Q(f(b)) *and* P(g(a,c))
Q(f(b)) *and* P(g(b,c))
Q(f(c)) *and* P(g(a,b))
Q(f(c)) *and* P(g(a,c))
Q(f(c)) *and* P(g(b,c))

Finalmente:

P(f(a)) *and* Q(g(a,a)
P(f(a)) *and* Q(g(b,b)
P(f(a)) *and* Q(g(c,c)

P(f(b)) *and* Q(g(a,a)
P(f(b)) *and* Q(g(b,b)
P(f(b)) *and* Q(g(c,c)
P(f(c)) *and* Q(g(a,a)
P(f(c)) *and* Q(g(b,b)
P(f(c)) *and* Q(g(c,c)
P(g(a,a)) *and* Q(f(a))
P(g(a,a)) *and* Q(g(a))
P(g(a,a)) *and* Q(g(a))
P(g(b,b)) *and* Q(g(b))
P(g(b,b)) *and* Q(g(b))
P(g(b,b)) *and* Q(g(b))
P(g(c,c)) *and* Q(g(c))
P(g(c,c)) *and* Q(g(c))
P(g(c,c)) *and* Q(g(c))
Q(f(a)) *and* P(g(b,a))
Q(f(a)) *and* P(g(c,a))
Q(f(a)) *and* P(g(c,b))
Q(f(b)) *and* P(g(b,a))
Q(f(b)) *and* P(g(c,a))
Q(f(b)) *and* P(g(c,b))
Q(f(c)) *and* P(g(b,a))
Q(f(c)) *and* P(g(c,a))
Q(f(c)) *and* P(g(c,b))

d)

```
p(f(X)) :- q(X, Y), r(Y).
q(g(X, Y), Z) :- r(X), r(Z), q(f(Z), a).
q(X, a).
r(f(f(b))).
r(c).
```

No se puede encontrar un modelo para el programa dado que en la primera fórmula, ya que el intŕprete al evaluar q(X,Y), dado que Y no esta unificado con ningún valor, él verifica que exista un predicado emphq con dos parámetros, como de hecho existe, y por consiguiente va a intentar unificar Y con a. Luego cuando evalue r(a) falla porque r(a) no existe como predicado verdadero.

e) r(f(f(b)) y r(c).

f) q(X,a).

g) H0 = Todo predicado en el conjunto P de predicados no debe tener variables libres, ni antecedentes. En otras palabras, deben ser hechos.
H1 = Todo predicado en el conjunto P de predicados puede tener variables libres, pero deben ser hechos.
Hk = Por cada predicado en el consecuente de cada conjunto de predicados, aplicar recursivamente la formula. K = min k1,k2,kp donde p es el numero de predicados en el consecuente.

i) El nivel de q es 0. Si se toma el nivel de los not en p y en r como 0, entonces ambos estaán a nivel 1. Si se toma al 1, entonces ambos estan al nivel 2.

j) Para el nivel 0:
Hk(q(X,Y) :- r(X), p(Y)) = Hk(r(X)) U Hk(p(y))
Hk(r(X)) = Hk-1(not(q(X,b)))
Hk(p(y)) = HK(q(X,Y) U Hk-1(not(q(X,X))) U Hk-1(not(r(Y)))

Para el nivel 1:
Hk(p(X)) U Hk(r(X))
Hk(p(X)) = Hk(q(X,Y) U Hk-1(not(q(X,X))) U Hk-1(not(r(Y))) Hk(r(X)) = Hk-1(not(q(X,b)))
k) Esto se ve afectado debido a que va hacer la recursion de manera infinita.