```
-----------------------------------------------
Teoría de Modelos. Clase del 1 de abril de 2020
-----------------------------------------------

---------------------------------------
Algunas propiedades de los programas ASP
---------------------------------------
([GK] páginas 47-51)
---------------------------------------

Hemos visto que, en general, un programa ASP puede tener un único
conjunto de respuestas, varios conjuntos de respuestas o ninguno.

-------------------------------------------------------------------
Problema: ¿Bajo qué condiciones podemos asegurar que un programa ASP
es *consistente* (esto es, tiene algún conjunto de respuestas)?
------------------------------------------------------------------

---------------------------------------------------
CASO 1: Programas positivos
---------------------------------------------------

-----------------------------------------------------------------------
Definición 1. Diremos que un programa ASP es *positivo* si no contiene
ni negación por defecto, ni negación clásica ni restricciones.

Proposición 1. Sea P un programa positivo. Entonces:
 a) P siempre tiene algún conjunto de respuestas.
 b) Si, además, P no contiene disyunciones, P tiene exactamente un
    conjunto de respuestas (que coincide con el menor modelo de Herbrand
de P).


--------------------------------------------------------------------
Ejemplo 1: Sea P1 el programa:
---------------------------------------------------------------------
bloque(a).
bloque(b).
bloque(c).
sobre(a,b).
sobre(b,c).
rojo(X) ; verde(X) ; azul(X):- bloque(X).
rojo(X)  :- sobre(X,Y),verde(Y).  %Sobre un bloque verde hay uno rojo.
verde(X) :- sobre(X,Y),rojo(Y).   %Sobre un bloque rojo hay uno verde.
azul(X)  :- sobre(X,Y),azul(Y).   %Sobre un bloque azul hay uno azul.
--------------------------------------------------------------------

Es claro que P1 es un programa positivo y, por tanto, es consistente.
```

Puesto que P1
contiene la disyunción epistémica(;), no podemos asegurar que tenga un
único
conjunto de respuestas. De hecho, P1 posee tres conjuntos de respuestas:

```
Answer: 1
bloque(a) bloque(b) bloque(c) sobre(a,b) sobre(b,c) verde(a) rojo(b)
verde(c)
Answer: 2
bloque(a) bloque(b) bloque(c) sobre(a,b) sobre(b,c) azul(a) azul(b)
azul(c)
Answer: 3
bloque(a) bloque(b) bloque(c) sobre(a,b) sobre(b,c) rojo(a) verde(b)
rojo(c)
```

Ejemplo 2: Sea P2 el programa:

```
r(a,b).
r(b,c).
r(Y,X) :- r(X,Y).
r(X,Z) :- r(X,Y),r(Y,Z).
```

Es claro que P2 es un programa positivo y, además, no contiene la
disyunción epistémica. Por tanto, P2 es consistente y tiene un único
conjunto
de respuestas (su menor modelo de Herbrand; el cual, en este caso,
coincide con el cierre simétrico y transitivo de la relación binaria r):

```
Answer: 1
r(a,b) r(b,c) r(c,b) r(b,a) r(a,c) r(c,a) r(a,a) r(b,b) r(c,c)
```

CASO 2: Programas normales

Definición 2. Diremos que un programa ASP es *normal* si no contiene
ni negación clásica ni restricciones (pero sí puede contener negación
por defecto *not*).

Nota: No es cierto que todo programa normal sea consistente. Basta
considerar
un programa tan simple como el dado por la regla:

```
p(a) :-not p(a).
```

Dicho programa es normal (no usa negación clásica ni restricciones) y, sin embargo, no posee conjuntos de respuestas. Hemos de considerar pues algún tipo particular de programas normales.

Definición 3. Sea P una programa normal sin variables. Una *función de nivel*
para P es una aplicación || || que asocia a cada átomo sin variables que aparece en
el programa P un número natural.

Ejemplo 3: Sea P3 el programa normal siguiente:

```
p(a) ; p(c) :- not p(b).
p(c) :- p(d), not p(a).
p(d).
```

Una función de nivel para P3 es, por ejemplo:
||p(a)||=0 ||p(b)|| =2 ||p(c)||=1 ||p(d)||=2

Nota: Una función de nivel se extiende de manera canónica a una disyunción de átomos
D = p1 or ... or pi tomando ||D|| = min{||p1||,...,||pi||}.
Por ejemplo, en el Ejemplo 3 anterior,
   ||p(a);p(c)||=min{||p(a)||,||p(c)||}= min{0,1} = 0

Definición 4. Sea P un programa normal sin variables. P se dirá *localmente estratificado*
si existe alguna función de nivel para P, || ||, tal que para toda regla R en P

```
   p_0 ; ... ; p_i :- p_i+1, ... ,p_m, not p_m+1, ..., not p_n.
       Cabeza(R)                    Cuerpo(R)
```

se cumple que:
  a) para cada p_k con i+1 <= k <=m (átomos en el cuerpo que no están afectados de *not*),
                ||p_k|| <= ||Cabeza(r)||;
  b) para cada p_k con m+1 <= k <=n (átomos en el cuerpo que sí están afectados de *not*),
                ||p_k|| < ||Cabeza(r)||.

Ejemplo 3(bis)

La función de nivel dada en el Ejemplo 3 anterior *no* nos permite probar que P3
es localmente estratificado. En efecto, la definición falla para la regla
  p(a) ; p(c) :- not p(b)
al no cumplirse que 2=||p(b)|| < ||p(a);p(c)||=0

Sin embargo, P3 sí es localmente estratificado. Para verlo, basta considerar una función de
nivel adecuada; por ejemplo:
  ||p(a)||= 1 ||p(b)||=0 ||p(c)||=2 ||p(d)||=2
Entonces:
  0=||p(b)|| < ||p(a);p(c)||=min{1,2}=1
  2=||p(d)|| <= ||p(c)||=2
  1=||p(a)|| < ||p(c)||= 2

Proposición 2. Sea P un programa normal sin variables.
 a) Si P está localmente estratificado, entonces P es consistente.
 b) Si P está localmente estratificado y no contiene la disyunción
    epistémica, entonces P tiene un único conjunto de respuestas.

Ejemplo 3(tris)

Puesto que el programa P3 está localmente estratificado, sabemos que
P3 es consistente. Puesto que P3 contiene disyunciones, no podemos asegurar
que tenga un único conjunto de respuestas. De hecho, P3 tiene dos
conjuntos de respuestas:

```
Answer: 1
p(d) p(a)
Answer: 2
p(d) p(c)
```

Ejemplo 4: Sea P4 el programa

```
seta(a).
seta(b).
gris(a) :- not rojo(a).
rojo(b) :- not gris(b).
venenosa(X) :- seta(X),rojo(X).
comestible(X) :- seta(X), not venenosa(X).
```

Es claro que P4 es un programa normal. Para aplicar la Proposición 2,
debemos calcular primero ground(P4).

```
ground(P4):
-----------
seta(a).                                          %H1
seta(b).                                          %H2
gris(a) :- not rojo(a).                           %R1
rojo(b) :- not gris(b).                           %R2
venenosa(a) :- seta(a),rojo(a).                   %R3
venenosa(b) :- seta(b),rojo(b).                   %R4
comestible(a) :- seta(a), not venenosa(a).        %R5
comestible(b) :- seta(b), not venenosa(b).        %R6
```

Demos una función de nivel para ground(P4) que muestre que P4 es localmente estratificado.

```
Átomos(P4) = {seta(a),seta(b),rojo(a),rojo(b),gris(a),gris(b),
              venenosa(a),venenosa(b),comestible(a),comestible(b)}
```

Cálculo de la función de nivel:

```
||seta(a)||=v1     ||seta(b)||=v2    ||rojo(a)||=v3      ||rojo(b)||=v4
||gris(a)||=v5     ||gris(b)||=v6    ||venenosa(a)||=v7  ||venenosa(b)||=v8
||comestible(a)||=v9       ||comestible(b)||=v10
```

Obtenemos un sistema de inecuaciones aplicando la Definición 4:

```
Por R1:   v3 < v5
Por R2:   v6 < v4
Por R3:   v1 <= v7  &&  v3 < v7
Por R4:   v2 <= v8  &&  v4 < v8
Por R5:   v1 <= v9  &&  v7 < v9
Por R6:   v2 <= v10 &&  v8 < v10
```

Una solución de este sistema de inecuaciones viene dado, por ejemplo, por:

```
  v1=v2=0  (ni seta(a) ni seta(b) aparecen en la cabeza de ninguna regla
que no sea un hecho)
  v3=v6=0  (ni rojo(a) ni gris(b) aparecen en la cabeza de ninguna regla)
  v4=v5=1  (para satisfacer R1 y R2)
  v7=v8=2  (para satisfacer R3 y R4)
  v9=v10=3 (para satisfacer R5 y R6)
```

Luego, la siguiente función de nivel para P4

```
||seta(a)||=0     ||seta(b)||=0    ||rojo(a)||=0      ||rojo(b)||=1
||gris(a)||=1     ||gris(b)||=0    ||venenosa(a)||=2  ||venenosa(b)||=2
||comestible(a)||=3     ||comestible(b)||=3
```

nos permite demostrar que P4 es localmente estratificado. Puesto que en P4 no
aparece la disyunción epistémica(;), por la Proposición 2 podemos inferir que
P4 es consistente y tiene un único conjunto de respuestas.
De hecho, dicho conjunto de respuestas viene dado por:

```
Answer: 1
rojo(b) gris(a) seta(a) seta(b) venenosa(b) comestible(a)
```

FIN DE LA CLASE
