

REPRESENTACIÓN DEL CONOCIMIENTO Y RAZONAMIENTO AUTOMÁTICO
 Examen 5/7/2019. Apellidos, nombre:

1a) (2 pts.) Dado el siguiente programa lógico proposicional P :

$$\begin{aligned} p &\leftarrow \text{not } q \\ q &\leftarrow \text{not } r \\ r &\leftarrow \text{not } p \end{aligned}$$

Indica cuáles son sus modelos clásicos mediante una tabla de verdad.

p	q	r	
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

1b) (3 pts.) Para cada modelo clásico I obtenido anteriormente, obtén el programa reducto P^I correspondiente, su modelo mínimo y, finalmente, indica si I es modelo estable (*stable model*). Usa tantas filas como precises.

modelo clásico I	programa reducto P^I	modelo mínimo de P^I	¿es estable? (sí/no)

- 2) En la lista de invitados de una boda, la familia nos facilita una lista con hechos para el predicado $odia(X, Y)$ donde X e Y son personas numeradas de 1 a $n = m * c$ y m es el número de mesas con capacidad para c personas cada una. Queremos obtener asignaciones a mesas en las que la gente no coincida con ninguna persona a la que odie. Las soluciones deben expresarse en términos del predicado $sienta(X, M)$, que sienta a la persona X en la mesa M .
- 2a) (3 pts.) Nos proporcionan el siguiente código ASP con algunos datos de ejemplo y la restricción principal. Completa el código para resolver correctamente el problema:

```
#const m=3.
#const c=3.
#const n=m*c.
persona(1..n).
mesa(1..m).
odia(3,5).
odia(1,9).
odia(8,9).
odia(2,4).
:- sienta(X,M), sienta(Y,M), odia(X,Y).          % (*)
```

- 2b) (1 pt.) Explica al menos dos situaciones en los datos de entrada que provocarían que el problema no tenga solución.
- 2c) (1 pt.) Dados los hechos de entrada y la restricción (*) que se proporciona en el apartado 2a) ¿Cuántos casos *ground* (esto es, sin variables) generará la regla (*)? Razona la respuesta.