

REPRESENTACIÓN DEL CONOCIMIENTO Y RAZONAMIENTO AUTOMÁTICO

Examen Parte I: Razonamiento Lógico 24 de mayo de 2023

Apellidos: _____ Nombre:_____

- **Puntuación:** esta examen de la parte I supone un tercio del total de la teoría que, a su vez, es el 40 % de la asignatura. En total, el máximo a obtener es $4/3 = \mathbf{1,33 \text{ puntos}}$ en el total de la asignatura. La puntuación de cada apartado del examen se mide en porcentaje de ese valor.
- Utilizar preferentemente el espacio reservado en el enunciado. En caso de equivocación, solicitar un formulario nuevo.

Ejercicio 1 (20 %). Dado el siguiente programa lógico P

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

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	×

Ejercicio 2 (30 %). Para cada uno I de los modelos clásicos obtenidos anteriormente, indica cuál es el reducto del programa anterior P^I , su modelo mínimo e indica si es o no un modelo estable. Usa tantas filas como necesites

modelo clásico I	programa reducto P^I	modelo mínimo de P^I	¿es estable? (sí/no)
$\{r\}$	$p \leftarrow q$ $r \leftarrow$	$\{r\}$	sí
$\{p, q\}$	$q \leftarrow p$	\emptyset	no
$\{p, r\}$	$r \leftarrow$	$\{r\}$	no
$\{p, q, r\}$		\emptyset	no

Ejercicio 3 (10 %).

a) Explica en qué consiste el criterio de *tolerancia a la elaboración* en un lenguaje de representación.

Una representación formal es tolerante a la elaboración en la medida que pequeños cambios en el problema a representar impliquen pequeños cambios en la representación de dicho problema. En otras palabras, cuando no hay tolerancia a la elaboración, puede ocurrir que un cambio pequeño en el problema implique rehacer toda la representación del mismo.

b) Asigna 1=mayor tolerancia, 2=intermedio, 3=menor tolerancia a los tres siguientes casos

Lenguaje natural = 1 Answer Set Programming = 2 Lógica Clásica = 3

Razona brevemente la ordenación que propones.

Aunque depende un poco del idioma que se use, en general, el lenguaje natural es probablemente el más tolerante a la elaboración que existe ya que es muy flexible y permite incorporar muchos matices y variantes de un problema con muy pocas palabras (su desventaja es que es muy ambiguo, pero esa es otra propiedad diferente). Entre ASP y la lógica clásica, la principal diferencia es que ASP permite razonamiento no monótono lo que hace posible extraer conclusiones en ausencia de información, lo que lo hace más flexible por ejemplo para realizar razonamiento por defecto. Es cierto que la lógica de predicados permite un manejo de cuantificadores más general que ASP, pero es una lógica monótona: no permite retractar una conclusión ante nueva evidencia.

Ejercicio 4 (10 %). Tenemos un predicado `producto(P)` que indica que P es un producto, un predicado `compra(X,P)` que indica que la persona X compra el producto P y un predicado `caro(P)` que dice que un producto es caro. Escribe una regla ASP que recoja en el predicado `favorito(P)` todos los productos P que compren al menos 3 personas y que no son caros.

```
favorito(P) :- producto(P), not caro(P), #count{X:compra(X,P)}>=3.
```

Ejercicio 5 (30 %). Un jugador lanza tres dados: uno azul, uno rojo y uno verde. Usamos el predicado `cara(D,V)` para indicar que el dado D muestra el valor V en su cara superior. Escribe un programa en ASP que genere todas las posibles combinaciones en las que los tres dados **no** forman valores consecutivos.

```
valor(1..6). dado(azul;rojo;verde).
1 {cara(D,V):valor(V)} 1 :- dado(D).
:- cara(X,V), cara(Y,V+1), cara(Z,V+2).
#show cara/2.
```

NOTA: los ejercicios 4 y 5 muestran dos posibles soluciones. Obviamente, no son las únicas formas correctas de codificar ambos problemas.