

Normas:

- La duración del examen es de **1 hora 40 minutos**
- Entregue los ejercicios en hojas separadas, incluidos en blanco
- No se permiten móviles, libros, apuntes, ni calculadora
- No está permitido levantarse del asiento salvo para entregar el examen.
- Al entregar el examen muestre el DNI al profesor

Ejercicio 1: [2 puntos]

Dado el siguiente modelo descrito en el formato Turtle, indicar las triplas que se deducen al hacer inferencia RDFS/OWL.

```
@prefix ex: <http://examen.urjc.es/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
```

```
ex:X rdf:type owl:Class .
ex:Y rdf:type owl:Class .
ex:Z rdf:type owl:Class .
ex:R rdf:type owl:Class .
ex:S rdf:type owl:Class .
ex:T rdf:type owl:Class .
ex:U rdf:type owl:Class .
```

```
ex:Z rdfs:subClassOf ex:X .
ex:R rdfs:subClassOf ex:Y .
ex:S rdfs:subClassOf ex:U .
ex:U rdfs:subClassOf ex:T .
```

```
ex:p rdf:type owl:ObjectProperty ;
     rdfs:domain ex:X ;
     rdfs:range ex:R .
```

```
ex:q rdf:type owl:ObjectProperty;
     rdf:type owl:TransitiveProperty .
```

```
ex:c rdf:type ex:S .
```

```
ex:a ex:p ex:b .
ex:c ex:q ex:d .
ex:d ex:q ex:e .
```

SOLUCIÓN

```
ex:a rdf:type ex:X .
```

```
ex:b rdf:type ex:R .
ex:b rdf:type ex:Y .
```

```
ex:p rdfs:range ex:Y .
```

```
ex:c rdf:type ex:U .
ex:c rdf:type ex:T .
```

```
ex:S rdfs:subClassOf ex:T .
ex:c ex:q ex:e .
```

Ejercicio 2: [2 puntos]

Dada la ontología que se muestra a continuación:

```
@prefix ex: <http://examen.org/vocab#> .
@prefix prop: <http://examen.org/property#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

# Definición de clases
ex:SoccerClub a rdfs:Class .

ex:SoccerPlayer a rdfs:Class .

ex:Stadium a rdfs:Class .

# Propiedades
prop:team a rdf:Property ;
    rdfs:domain ex:SoccerPlayer ;
    rdfs:range ex:SoccerClub ;
    rdfs:comment "Indica el equipo para el que juega un jugador." .

prop:totalGoals a rdf:Property ;
    rdfs:domain ex:SoccerPlayer ;
    rdfs:range xsd:integer ;
    rdfs:comment "Número total de goles anotados por un jugador." .

prop:homeStadium a rdf:Property ;
    rdfs:domain ex:SoccerClub ;
    rdfs:range ex:Stadium ;
    rdfs:comment "Estadio donde juega el equipo." .

prop:seatingCapacity a rdf:Property ;
    rdfs:domain ex:Stadium ;
    rdfs:range xsd:integer ;
    rdfs:comment "Capacidad de asientos del estadio." .
```

Se pide:

Realizar una consulta SPARQL para obtener los equipos de fútbol en los que hayan jugado al menos dos jugadores que hayan marcado más de 190 goles y que tengan un estadio con una capacidad para más de 90000 espectadores. Los resultados deben mostrarse ordenados por equipo.

NOTA: No es necesario definir los prefijos ya incluidos en el enunciado.

SOLUCIÓN

```
PREFIX ex: <http://examen.org/vocab#>
PREFIX prop: <http://examen.org/property#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT DISTINCT ?equipo
WHERE {
    # Aseguramos que la entidad es un club de fútbol
    ?equipo rdf:type ex:SoccerClub.

    # Identificamos dos jugadores diferentes
    ?jugador1 rdf:type ex:SoccerPlayer.
    ?jugador2 rdf:type ex:SoccerPlayer.
    #Aseguramos que los jugadores sean diferentes
    FILTER (?jugador1 != ?jugador2)

    # Ambos jugadores deben pertenecer al mismo equipo
    ?jugador1 prop:team ?equipo.
    ?jugador2 prop:team ?equipo.

    # Comprobamos que cada jugador debe haber anotado más de 190
    ?jugador1 prop:totalGoals ?goles1.
    FILTER (?goles1 > 190)
    ?jugador2 prop:totalGoals ?goles2.
    FILTER (?goles2 > 190)

    # Verificamos que el estadio tiene más de 90,000 asientos
    ?equipo prop:homeStadium ?estadio.
    ?estadio prop:seatingCapacity ?capacidad.
    FILTER (?capacidad > 90000)
}
ORDER BY ?equipo
```

Ejercicio 3: [3 puntos]

Dados los siguientes nombres de concepto: *Teatro*, *Estadio*, *Recinto*, *Asiento*, *Puerta*, *Ciudad*, *Aburrida*, *Lugar* y *Artista*, y los nombres de rol *tiene* y *actúa*, representar el siguiente conocimiento en lógica de descripciones \mathcal{ALC} y en lógica de primer orden.

- 1) Los teatros y los estadios son recintos
- 2) Los estadios y los teatros son cosas diferentes
- 3) Los estadios tienen asientos y puertas
- 4) Las ciudades que tienen teatro y estadio no son aburridas
- 5) Los teatros son lugares donde sólo actúan artistas
- 6) Madrid es una ciudad que tiene el estadio Santiago Bernabéu y también tiene teatros

Solución:

- 1) Los teatros y los estadios son recintos

$\text{Teatro} \sqcup \text{Estadio} \sqsubseteq \text{Recinto}$
 $\forall x(\text{Teatro}(x) \vee \text{Estadio}(x) \rightarrow \text{Recinto}(x))$

- 2) Los estadios y los teatros son cosas diferentes

$\text{Estadio} \sqsubseteq \neg \text{Teatro}$
 $\forall x(\text{Estadio}(x) \rightarrow \neg \text{Teatro}(x))$

- 3) Los estadios tienen asientos y puertas

$\text{Estadio} \sqsubseteq \exists \text{tiene.Asiento} \sqcap \exists \text{tiene.Puerta}$
 $\forall x(\text{Estadio}(x) \rightarrow \exists y(\text{tiene}(x,y) \wedge \text{Asiento}(y)) \wedge \exists z(\text{tiene}(x,z) \wedge \text{Puerta}(z)))$

- 4) Las ciudades que tienen teatro y estadio no son aburridas

$\text{Ciudad} \sqcap \exists \text{tiene.Teatro} \sqcap \exists \text{tiene.Estadio} \sqsubseteq \neg \text{Aburrida}$
 $\forall x(\text{Ciudad}(x) \wedge \exists y(\text{tiene}(x,y) \wedge \text{Teatro}(y)) \wedge \exists z(\text{tiene}(x,z) \wedge \text{Estadio}(z)) \rightarrow \neg \text{Aburrida}(x))$

- 5) Los teatros son lugares donde sólo actúan artistas

$\text{Teatro} \sqsubseteq \text{Lugar} \sqcap \forall \text{actúa.Artista}$
 $\forall x(\text{Teatro}(x) \rightarrow \text{Lugar}(x) \wedge \forall y(\text{actúa}(x,y) \rightarrow \text{Artista}(y)))$

- 6) Madrid es una ciudad que tiene el estadio Santiago Bernabéu y también tiene teatros

$\{\text{Ciudad}(\text{Madrid}), \text{Estadio}(\text{Bernabéu}), \text{tiene}(\text{Madrid}, \text{Bernabéu}), \exists \text{tiene.Teatro}(\text{Madrid})\}$

$\text{Ciudad}(\text{Madrid}) \wedge \text{Estadio}(\text{Bernabéu}) \wedge \text{tiene}(\text{Madrid}, \text{Bernabéu}) \wedge$
 $\wedge \exists x(\text{tiene}(\text{Madrid}, x) \wedge \text{Teatro}(x))$

Ejercicio 4: [3 puntos]

Se tiene una compuerta que forma parte del sistema de retención de agua del cauce de un río. La compuerta está cerrada y se eleva cuando se enciende el sistema. Durante los primeros 10 minutos la apertura es del 50% (*apertura_50*) y luego se amplía al 100% (*apertura_100*) hasta cerrar la compuerta (la compuerta no puede estar abierta al 50% y al 100% simultáneamente). La compuerta se acciona manualmente y en $T = 0$ está cerrada. Utilice Basic Event Calculus (BEC) para modelar las siguientes acciones:

- La compuerta se abre en $T = 8$.
- La compuerta se cierra en $T = 40$.

Considere la siguiente representación de los fluents y events:

Fluents	Events
<i>Comp_abierta.</i>	<i>abrir_Comp.</i>
<i>apertura_50.</i>	<i>cerrar_Comp.</i>
<i>apertura_100.</i>	

Bajo estas consideraciones se solicita:

- Definir el modelo (1 punto) y las acciones (0.5 pts) de acuerdo con la narrativa anterior.
- Inferir si la compuerta está abierta al 100% $T = 20$ (1.5 pts).

Utilice los axiomas del BEC:

Predicate	Meaning
<i>InitiallyN(f)</i>	<i>f</i> is false at timepoint 0
<i>InitiallyP(f)</i>	<i>f</i> is true at timepoint 0
<i>HoldsAt(f, t)</i>	<i>f</i> is true at <i>t</i>
<i>Happens(e, t)</i>	<i>e</i> occurs at <i>t</i>
<i>Initiates(e, f, t)</i>	if <i>e</i> occurs at <i>t</i> , then <i>f</i> is true and not released from the commonsense law of inertia after <i>t</i>
<i>Terminates(e, f, t)</i>	if <i>e</i> occurs at <i>t</i> , then <i>f</i> is false and not released from the commonsense law of inertia after <i>t</i>
<i>Releases(e, f, t)</i>	if <i>e</i> occurs at <i>t</i> , then <i>f</i> is released from the commonsense law of inertia after <i>t</i>
<i>StoppedIn(t₁, f, t₂)</i>	<i>f</i> is stopped between <i>t₁</i> and <i>t₂</i>
<i>StartedIn(t₁, f, t₂)</i>	<i>f</i> is started between <i>t₁</i> and <i>t₂</i>
<i>Trajectory(f₁, t₁, f₂, t₂)</i>	if <i>f₁</i> is initiated by an event that occurs at <i>t₁</i> , then <i>f₂</i> is true at <i>t₁ + t₂</i>

- BEC1. $StoppedIn(t_1, f, t_2) \equiv \exists e, t (Happens(e, t) \wedge t_1 < t < t_2 \wedge (Terminates(e, f, t) \vee Releases(e, f, t)))$.
- BEC2. $StartedIn(t_1, f, t_2) \equiv \exists e, t (Happens(e, t) \wedge t_1 < t < t_2 \wedge (Initiates(e, f, t) \vee Releases(e, f, t)))$.
- BEC3. $Happens(e, t_1) \wedge Initiates(e, f_1, t_1) \wedge 0 < t_2 \wedge Trajectory(f_1, t_1, f_2, t_2) \wedge \neg StoppedIn(t_1, f_1, t_1 + t_2) \supset HoldsAt(f_2, t_1 + t_2)$.
- BEC4. $InitiallyP(f) \wedge \neg StoppedIn(0, f, t) \supset HoldsAt(f, t)$.
- BEC5. $InitiallyN(f) \wedge \neg StartedIn(0, f, t) \supset \neg HoldsAt(f, t)$.
- BEC6. $Happens(e, t_1) \wedge Initiates(e, f, t_1) \wedge t_1 < t_2 \wedge \neg StoppedIn(t_1, f, t_2) \supset HoldsAt(f, t_2)$.
- BEC7. $Happens(e, t_1) \wedge Terminates(e, f, t_1) \wedge t_1 < t_2 \wedge \neg StartedIn(t_1, f, t_2) \supset \neg HoldsAt(f, t_2)$.

Let BEC be the conjunction of BEC1 through BEC7.

Solución

<p>a) Modelo</p> <p><i>InitiallyN(Comp_abierta).</i> $\neg(\text{HoldsAt}(\text{apertura_50}, T) \wedge \text{HoldsAt}(\text{apertura_100}, T)).$ <i>Initiates(abrir_Comp, Comp_abierta, T).</i> <i>Terminates(cerrar_Comp, Comp_abierta, T).</i> <i>Releases(abrir_Comp, apertura_50, T).</i> <i>Releases(abrir_Comp, apertura_100, T).</i> <i>Trajectory(Comp_abierta, T1, apertura_50, T2) $\leftarrow 0 < T2 \leq 10$.</i> <i>Trajectory(Comp_abierta, T1, apertura_100, T2) $\leftarrow T2 > 10$.</i></p>	<p>Acciones</p> <p><i>Happens(abrir_Comp, 8).</i> <i>Happens(cerrar_Comp, 40)</i> .</p>
<p>b) Inferencia</p> <p>? <i>HoldsAt(apertura_100, 20).</i></p> <ul style="list-style-type: none"> - Por <i>Happens(abrir_Comp, 8)</i>, en BEC1 tenemos que $\neg \text{StoppedIn}(8, \text{Comp_abierto}, 20)$ dado que no existe ningún evento entre 8 y 20 (<i>Happens(e, T)</i> con $8 < T \leq 20$) que cierre la compuerta y, por ende, no existe <i>Terminates(e, Comp_abierta, T)</i> en ese lapso tiempo. Además, no existe <i>Releases(e, Comp_abierta, T)</i>. - La diferencia en tiempo entre el evento <i>Happens(abrir_compuerta, 8)</i> y la consulta <i>HoldsAt(apertura_100, 20)</i> es necesaria para determinar cuál <i>Trajectory</i> (<i>Trajectory(Comp_abierta, T1, apertura_50, T2)</i> o <i>Trajectory(Comp_abierta, T1, apertura_100, T2)</i>) es verdadera. Dado que $T1 + T2 = 20$, es decir, $8 + T2 = 20$, tenemos que $T2 = 12$. Como $T2 = 12$, entonces <i>Trajectory(Comp_abierta, 8, apertura_100, 12)</i> es verdadera. - Por BEC 3 tenemos que <i>HoldsAt(apertura_100, 20)</i> es verdadero. Es decir: <i>Happens(abrir_Comp, 8)</i> e <i>Initiates(abrir_Comp, Comp_abierta, 8)</i> y $0 < 12$ y <i>Trajectory(Comp_abierta, 8, apertura_100, 12)</i> y $\neg \text{StoppedIn}(8, \text{Comp_abierto}, 20)$ entonces <i>HoldsAt(apertura_100, 20)</i>. 	