

<Spoiler Alert>



"¿Te dije que House of Cards no va más? Seguime a mí, pichón..."

Continuamos con el desarrollo de la aplicación de Spoilers en Prolog. Ahora nos han pedido nuevos requerimientos que describiremos a continuación.

1 Punto A: Malo, malo, malo eres...

Desarrollar el predicado **malaGente/1** que se cumple para una persona si a todas las personas con las que habló les spoileó algo, o si a alguien le spoileó una serie que la mala persona ni siquiera mira.

Incorporemos a la base de conocimiento que Nico le dice a Juan acerca de la muerte de Seymour Diera en Futurama.

También sumemos a Pedro, que mira Game of Thrones.

Pedro le dijo a Aye que en la serie Game of Thrones hay una relación de amistad entre Tyrion y el dragón (que es cierta). Recordamos que Aye planea ver Game of Thrones.

También le dijo a Nico que hay una relación de parentesco en Game of Thrones entre Tyrion y el dragón, cosa que no es cierta.

- Gastón es mala gente, porque le spoileó a todas las personas a las que le habló (Maiu).
- Nico es mala gente, porque le spoileó a Juan la muerte de Seymour Diera en Futurama, serie que Nico ni siquiera mira.
- Pedro no es mala gente, porque si bien le spoileó a Aye, le habló a Nico sin spoilearle.

2 Punto B: Series con cosas fuertes

Queremos agregar nuevas cosas que pasaron en la serie. Hasta el momento podía pasar que un personaje muera o que se revele una relación de una determinada

naturaleza entre dos personas, ahora también queremos reflejar si ocurre un giro en la trama (plot twist), el cual se describe mediante una lista de palabras clave¹.

Debemos incorporar a la base los siguientes plot twists:

- En Game of Thrones, temporada 3, capítulo 2, aparecen las palabras sueño (suenio) y “sin piernas” (sinPiernas)
- En Game of Thrones, temporada 3, capítulo 12, aparecen las palabras fuego y boda
- En Supercampeones, temporada 9, capítulo 9, aparecen las palabras sueño, coma y sin piernas
- En Doctor House, temporada 8, capítulo 7, aparecen las palabras coma y pastillas

Se pide modificar el predicado que indica que algo es fuerte (de la entrega 1)², donde sabemos que en toda serie una muerte y una relación de parentesco o amorosa es algo fuerte.

En cuanto al giro o *plot twist*, es fuerte si no es cliché y pasó en un final de temporada. Consideramos que el giro es cliché si todas las palabras clave aparecen en giros de otras series (no necesariamente la misma serie para cada palabra).

Sabemos que

- la muerte de Seymour Diera en Futurama es algo fuerte
- también la muerte de Emperor en Star Wars es algo fuerte
- la relación de parentesco de Anakin y el Rey en Star Wars es algo fuerte
- la relación de parentesco de Darth Vader y Luke en Star Wars es algo fuerte
- la relación amorosa de Ted y Robin en How I met your mother es algo fuerte
- la relación amorosa de Swarley y Robin en How I met your mother es algo fuerte
- el plot twist que contiene las palabras fuego y boda en Game of Thrones (que no es cliché y pasó en un final de temporada) es algo fuerte
- el plot twist que contiene la palabra sueño en Game of Thrones no es fuerte, porque es un cliché (“sueño” aparece por ejemplo en Super Campeones)
- el plot twist que contiene las palabras coma y pastillas en Doctor House no es fuerte, ya que si bien no es cliché, no pasó en final de temporada

3 Punto C: Popularidad

¡¡Buenas noticias!! Se nos ocurrió una forma de generalizar cuándo una serie es popular. Modificar predicado **popular/1** considerando que se cumple para una serie si su popularidad, que es la cantidad de personas que la miran multiplicado por la cantidad de conversaciones de la misma, es mayor o igual que la conseguida por Star

¹ Recomendamos nombrar a esa característica plotTwist para facilitar el testeo unitario automatizado

² Recomendamos nombrar a este predicado simplemente **fuerte/2** para facilitar el testeo unitario automatizado

Wars, la cual consideramos como serie de referencia por su alta cantidad de fans en un grupo relativamente selecto.

Game of Thrones y Star Wars (obviamente) son populares.
Además House of Cards es popular, independientemente de si se habla o no de ella.

4 Punto D: Amigos son los amigos...

Agregamos a la base de conocimientos la relación de amistad entre dos personas:

```
amigo(nico, maiu).  
amigo(maiu, gaston).  
amigo(maiu, juan).  
amigo(juan, aye).
```

La relación **amigo/2** es unidireccional y cerrada.

Resolver el predicado **fullSpoil/2** que relaciona dos personas cuando el primero le dijo un spoil al segundo directamente o bien se lo dijo a un amigo del segundo, o al amigo de un amigo del segundo, o al amigo... en fin, creo que se entiende que la relación debe soportar n niveles de anidamiento. No es válido el *auto-full-spoil*.

Por ejemplo:

- Nico hizo full spoil a Aye, Juan, Maiu y Gastón
- Gastón hizo full spoil a Maiu, Juan y Aye
- Maiu no hizo full spoil a nadie (no le dijo nada a nadie)

5 Casos de prueba

A continuación proveemos la implementación de los casos de prueba unitarios automatizados que deben pasar exitosamente:

```
:- begin_tests(spoiler_alert).  
  
    % Test Entrega 1 - Punto B - Es spoiler  
    test(esSpoilerMuerteEmperorEnStarWars, nondet) :-  
        esSpoiler(starWars, muerte(emperor)).  
  
    test(esSpoilerMuertePedroEnStarWars, nondet) :-  
        not(esSpoiler(starWars, muerte(pedro))).  
  
    test(esSpoilerParentescoAnakinReyEnStarWars, nondet) :-  
        esSpoiler(starWars, relacion(parentesco, anakin, rey)).  
  
    test(esSpoilerParentescoAnakinLavezziEnStarWars, nondet) :-  
        not(esSpoiler(starWars, relacion(parentesco, anakin, lavezzi))).
```

```

% Test Entrega 1 - Punto C - Le spoileo
test(leSpoileoGastonAMaiugoT, nondet) :-
    leSpoileo(gaston, maiu, got).

test(leSpoileoNicoAMaiuStarWars, nondet) :-
    leSpoileo(nico, maiu, starWars).

% Test Entrega 1 - Punto D - Televidente responsable
test(nicoNoEsResponsable, nondet):-
    not(televidenteResponsable(nico)).

test(gastonNoEsResponsable, nondet):-
    not(televidenteResponsable(gaston)).

test(ayeSoloEsUnaTroll, nondet):-
    televidenteResponsable(aye).

test(televidenteResponsableEsInversible, set(Televidente == [juan,
maiu, aye])):-
    televidenteResponsable(Televidente).

% Test Entrega 1 - Punto E - Viene zafando
test(maiuNoVieneZafando, nondet):-
    not(vieneZafando(maiu, _)).
test(juanVieneZafando, set(Serie == [himym, got, hoc])):-
    vieneZafando(juan, Serie).
test(soloNicoVieneZafandoConStarWars, set(Persona == [nico])):-
    vieneZafando(Persona, starWars).

% Test Entrega 2

% Punto A - malaGente
test(malaGente, set(Malos = [nico, gaston])):-
    malaGente(Malos).

test(buenaGente, set(Buenos = [aye, maiu, pedro, juan])):-
    enganchado(Buenos, _), not(malaGente(Buenos)).

% Punto B - esFuerte
test(fuerte1, nondet):-
    fuerte(futurama, muerte(seymourDiera)).

test(fuerte2, nondet):-
    fuerte(starWars, muerte(emperor)).

test(fuerte3, nondet):-
    fuerte(starWars, relacion(parentesco, anakin, rey)).

```

```

test(fuerte4, nondet):-
    fuerte(starWars, relacion(parentesco, vader, luke)).

test(fuerte5, nondet):-
    fuerte(himym, relacion(amorosa, ted, robin)).

test(fuerte6, nondet):-
    fuerte(himym, relacion(amorosa, swarley, robin)).

test(fuerte7, nondet):-
    fuerte(got, plotTwist([fuego, boda])).

test(fuerte8, nondet):-
    not(fuerte(got, plotTwist([suenio]))).

test(fuerte9, nondet):-
    not(fuerte(drHouse, plotTwist([coma, pastillas]))).

% Punto C - popularidad
test(popularidad, set(Populares = [got, starWars, hoc])):-
    popular(Populares).

% Punto D - fullSpoil
test(fullSpoil, set(Quienes = [gaston, aye, juan, maiu])):-
    fullSpoil(nico, Quienes).

test(fullSpoil2, set(Quienes = [aye, juan, maiu])):-
    fullSpoil(gaston, Quienes).

test(fullSpoil3, set(Quienes = [])):-
    fullSpoil(maiu, Quienes).

:- end_tests(spoiler_alert).

```

Debe agregar estas líneas al final del apunte y sincronizar los nombres de los predicados que usted defina con los que piden los tests.

Para más información recomendamos leer el apunte [PLUnit para programadores prologueros](#).

6 Condiciones de entrega: IMPORTANTE leer para evitar malos entendidos

Cada persona debe

- entregar el archivo .pl con las definiciones de predicados

- incluyendo los tests adaptados para su trabajo práctico
- todos los tests deben pasar correctamente
- la fecha tope de entrega es el **11/07** a las **09:00** hs.

El no cumplimiento de alguno de estos puntos determinará que el tutor pueda considerar inválida la entrega del TP.

7 Conceptos a evaluar

- Modelado de información
- Principio de universo cerrado
- Cuantificación universal (para todos)
- Negación
- Orden superior
- Inversibilidad
- Recursividad