

<Spoiler Alert>

Lamentablemente, existe gente mala en el mundo, ese tipo de personas que spoilean una buena serie, ya sea por diversión o distracción. También existen los trolls, que son aquellos que dan falsos spoilers para arruinar la vida hasta que compruebes que fue todo una gran mentira.

Es por este motivo que hemos decidido realizar un programa en el Paradigma Lógico, en el lenguaje Prolog, para poder adentrarnos más en el mundo de los Spoilers y saber realmente cuándo debemos matar a alguien o no.



"...Porque el TP es largo, y está lleno de spoilers"

1 Punto A: Quién mira qué

Sabemos que

- Juan mira "How I met your mother", "Futurama" y "Game of Thrones"
- Nadie mira "Mad men"
- A Star Wars la miran Nico y Maiu
- Maiu ve "One Piece" y "Game of Thrones"
- Nico ve "Game of Thrones"
- Gastón está viendo "House of Cards"
- Alf no ve ninguna serie porque el doctorado le consume toda la vida

Además sabemos que las series "Game of Thrones", "House of Cards" y "Star Wars" son populares.

También queremos registrar las series que la gente planea ver. Por ejemplo:

- Juan quiere ver "House of Cards"
- Aye quiere ver "Game of Thrones"
- Gastón quiere ver "How I met your mother"

Por último queremos registrar de cada serie, cuántos episodios tiene cada temporada. Por ejemplo:

- "Game of Thrones" tiene 12 episodios la temporada 3 y 10 la segunda temporada.
- "How I met your mother" tiene 23 episodios la temporada 1
- "Dr. House" tiene 16 episodios la temporada 8
- No recordamos cuántos episodios tiene la segunda temporada de "Mad men"

Definir la base de conocimientos y en caso de no implementar nada justificar qué concepto entra en juego.

Nota: Para identificar a las series mencionadas anteriormente usar los siguientes nombres: himym, starWars, got, futurama, hoc, onePiece, drHouse, madMen.

2 Anexo: Lo que pasó, pasó

Registramos las cosas importantes que pasaron en distintas series, que puede ser:

- Un personaje muere
- Se devela una relación de una determinada naturaleza entre dos personas

En este caso proveemos la base de conocimientos que deben agregar:

```
%paso(Serie, Temporada, Episodio, Lo que paso)
paso(futurama, 2, 3, muerte(seymourDiera)).
paso(starWars, 10, 9, muerte(emperor)).
paso(starWars, 1, 2, relacion(parentesco, anakin, rey)).
paso(starWars, 3, 2, relacion(parentesco, vader, luke)).
paso(himym, 1, 1, relacion(amorosa, ted, robin)).
paso(himym, 4, 3, relacion(amorosa, swarley, robin)).
paso(got, 4, 5, relacion(amistad, tyrion, dragon)).
```

También queremos relacionar una persona con otra y algo que supuestamente pasó en una serie que la otra persona no sabía:

```
%leDijo/4
leDijo(gaston, maiu, got, relacion(amistad, tyrion, dragon)).
leDijo(nico, maiu, starWars, relacion(parentesco, vader, luke)).
leDijo(nico, juan, got, muerte(tyrion)).
leDijo(aye, juan, got, relacion(amistad, tyrion, john)).
leDijo(aye, maiu, got, relacion(amistad, tyrion, john)).
leDijo(aye, gaston, got, relacion(amistad, tyrion, dragon)).
```

3 Punto B: Es spoiler

Implementar el predicado **esSpoiler/2** que relaciona una serie y un posible spoiler, que se cumple si efectivamente pasó (claro, tiene que morir **ese** personaje, o darse **esa** relación entre **esos** personajes concretos).

Por ejemplo:

- es cierto que la muerte de Emperor para Star Wars es un spoiler.
- no es cierto que la muerte de Pedro sea un spoiler para Star Wars.
- es cierto que la relación de parentesco entre Anakin y el Rey es un spoiler para Star Wars
- no es cierto que la relación de padre entre Anakin y Lavezzi sea un spoiler para Star Wars

¿Qué tipo de consultas (existenciales o individuales) pueden hacerse a la base de conocimientos? Justificar a partir de conceptos vistos en la cursada.

4 Punto C: Te pedí que no me lo dijeras

Pedimos que resuelvas el predicado **leSpoileo/3** que relaciona a 2 personas y una serie si la primera le dijo a la segunda un spoiler para una serie que la segunda mira o planea ver.

Consejo: separar las responsabilidades en predicados más chicos.

Por ejemplo

- Gastón le dijo a Maiu un spoiler de Game of Thrones (la relación de amistad entre Tyrion y el dragón)
- Nico le dijo a Maiu un spoiler de Star Wars (la relación de parentesco entre Darth Vader y Luke Skywalker)

No se preocupen si en los resultados aparece más de una vez, esto es así por la naturaleza del motor de inferencia de Prolog.

¿Qué tipo de consultas (existenciales o individuales) pueden hacerse a la base de conocimientos? Justificar a partir de conceptos vistos en la cursada.

5 Punto D: Responsable

Resolver el predicado **televidenteResponsable/1** que se cumple para una persona si no le spoileó ninguna serie a nadie.

Por ejemplo:

- Juan, Aye y Maiu son televidentes responsables. ¿Pero Aye no le anduvo diciendo cosas a mucha gente? Sí, pero Aye le dijo a Maiu y a Nico cosas que nunca ocurrieron (no es spoiler), y Aye le dijo a Gastón algo que sí ocurrió, pero es sobre Game of Thrones, que Gastón no mira ni planea ver. Entonces es una televidente responsable según los criterios del negocio.
- Nico y Gastón no son televidentes responsables

BONUS: Lograr que el predicado sea inversible.

6 Punto E: Viene Zafando

Resolver el predicado **vieneZafando/2** que se cumple para alguien que mira o planea ver una serie y no se la spoilearon, a pesar de que la misma sea popular o que en todas sus temporadas hayan pasado cosas fuertes. Algo que pasó se considera fuerte si se trata de una muerte o de una relación amorosa o de parentesco. **Evitar duplicación de ideas.**

Por ejemplo:

- Maiu no viene zafando con ninguna serie.
- Juan viene zafando tanto con "How I Met Your Mother", "Game of Thrones" y "House of Cards". No viene zafando con "Futurama", porque no tenemos información sobre las temporadas que tiene (aunque sepamos que pasó algo fuerte no nos alcanza)
- Sólo Nico viene zafando con "Star Wars"

7 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,
maiui, 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).

:- 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](#).

8 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 **27/06** a las **09:00** hs.

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

9 Conceptos a evaluar

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