INF-253 Lenguajes de Programación
Tarea 5: Prolog

Profesor: José Luis Martí Lara, Roberto Diaz Urra
Ayudantes Cátedra: Gabriela Acuña Benito, Hugo Sepúlveda Arriaza,
Lucio Fondón Rebolledo
Ayudantes Tareas: Gabriel Carmona Tabja, Domingo Benoit Cea,
Héctor Larrañaga Jorquera, Ignacio Ulloa Huenul,
Joaquín Gatica Hernández, Javier Pérez Riveros,
José Runín Basáez, Rafael Aros Soto
Rodrigo Pérez Jamett

6 de junio de 2022

1. Mecánica

Se les entregarán dos problemas a resolver utilizando el lenguaje de programación Prolog.

Para poder realizar la tarea, se debe utilizar SWI-Prolog, el cual se puede encontrar en: https://www.swi-prolog.org/download/stable.

2. Gablema - Navidad Grupal

El informático anónimo esta ansioso por la navidad, debido a sus recuerdos de la infancia se dio cuenta que mientras el estaba con algún niño o niña que se haya portado mal nunca le llegó un regalo. Entonces, el informático anónimo se percató que el viejito pascuero mira grupos bondadosos y no el individualismo de las personas. Entonces, le mandó una carta emocionado al viejito indicándole su descubrimiento y el informático anónimo le comentó que podía prestarle ayuda desarrollando un programa de prolog. Sorprendentemente, el viejito pascuero le respondió que llevará su programa el polo norte. Es tu momento de brillar.

Su programa debe recibir una lista de listas, donde cada lista representa un grupo y contiene números de bondad (números enteros), luego sí es que el promedio del grupo es mayor a la mediana del grupo se considera un grupo bondadoso, sino se considera que hubo mucho individualismo y no se dará regalo. Su consulta por cada grupo debe mostrar verdadero o falso, donde verdadero es que es un grupo bondadoso y falso si no lo es.

```
?- bondad([[1, 2, 5], [1, 2, 3]], L).

L = [true, false]
```

3. Pantema - ¿Expresiones regulares?

El informático anónimo es nostálgico, siempre le gusta pensar en el pasado. Por esto, él se recuerda de su programa de python que permitía ejecutar el lenguaje MovMario, luego de pensar varias horas en su código piensa: "¿Podrá Prolog ejecutar operaciones?". Entonces, desarrolla una sintaxis que permitiría ejecutar operaciones matemáticas. Las operaciones matemáticas las cuales se pueden ejecutar son: suma ("+"), resta ("-"), multiplicación ("*"), división ("/"), modulo ("mod"), potenciación ("^"). Y se escribe utilizando lo que corresponde a un árbol sintáctico, representado como lista de listas.

Al final, este problema busca implementar la siguiente expresión regular:

Listing 1: EBNF

```
numero ::= 0 | no_zero {'0' | no_zero}

no_zero ::= '1' | '2' | '3' | '4' | '5' | '6' | '7' | '8' | '9'

operacion ::= (operacion | numero) ('+' | '-' | '*' | '/' | 'mod' | '^') (operacion | numero)
```

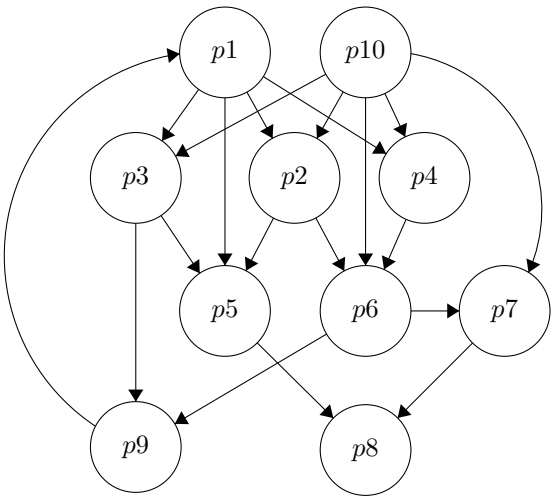
Entonces tiene que hacer un programa que reciba el árbol sintáctico y entrega el resultado de la operación.

```
?- matematica([2, "+", [[1, "-", 4], "^", 2]], RES).

RES = 11
```

4. GabPantema - La Ancestrosidad

El informático anónimo le gusta inventar conceptos porque sí. Aprovechando que se está acercando la navidad y tiene que empezar a pensar en los regalos de sus familiares, otra pregunta nace. ¿Cuan ancestro es una persona con respecto a otra? A este concepto, él lo llama la ancestrosidad entre dos personas. Por ejemplo, si se tiene el siguiente árbol familiar (corresponde a una familia misteriosa de antaño):



Si se pregunta por la ancestrosidad de $p9$ con $p1$, este dirá que es 2, esto debido que la distancia para ir del ancestro $p1$ hasta $p9$ es de 2. Ahora bien, si se pregunta por la ancestrosidad de $p1$ con $p9$, este dirá 1, esto debido que $p9$ tiene una conexión directa con $p1$.

A partir del ejemplo anterior, se puede notar que si yo hago la siguiente consulta $ancestrosidad(A, B)$ debe buscar la distancia mínima yendo desde B hasta A.

Dados los hechos de este árbol, que corresponde a una relación $ancestro(x, y)$ donde significa que y es ancestro de x, debe poder determinar la ancestrosidad de x con y. En el caso, de que B no pueda ser ancestro con A de alguna forma, el valor de ancestrosidad será -1. El árbol puede tener ciclos.

```
ancestro(p2, p1).
ancestro(p3, p1).
ancestro(p4, p1).
ancestro(p5, p1).

ancestro(p2, p10).
ancestro(p3, p10).
ancestro(p4, p10).
ancestro(p6, p10).
ancestro(p7, p10).

ancestro(p5, p2).
ancestro(p6, p2).

ancestro(p5, p3).
ancestro(p9, p3).

ancestro(p6, p4).

ancestro(p8, p5).

ancestro(p7, p6).
ancestro(p9, p6).

ancestro(p8, p7).

ancestro(p1, p9).
```

```
?- ancestrosidad(p9, p1, X)
X = 2
?- ancestrosidad(p1, p9, X)
X = 1
?- ancestrosidad(p1, p8, X)
X = -1
```

DATO IMPORTANTE: NO SE ACEPTARÁN RESPUESTAS QUE DEFINA TODAS LAS OPCIONES POSIBLES COMO HECHOS EN ESTE PROBLEMA.

5. Reglas del Juego

5.1. Archivos a entregar

- Pantema: pantema.pro o pantema.pl
- Gablema: gablema.pro o gablema.pl
- GabPantema: gabpantema.pro o gabpantema.pl

5.2. Otras Reglas

- Cuidado con el orden y la identación de su tarea, puede llevar descuentos.
- Las copias serán evaluadas con nota 0 y se informarán a las respectivas autoridades.
- El código debe ser correctamente comentado, si no habrá descuento. Me refiero a descripción de las reglas y hechos.
- **La entrega debe realizarse en tar.gz y debe llevar el nombre: Tarea5LP_RolIntegrante.tar.gz**
- La entrega debe incluir un README.txt con el nombre del estudiante e instrucciones para utilizar su programa.
- **La tarea debe ser entregada el 12 de Diciembre a las 23:59, tareas atrasadas por más de 3 horas no serán aceptadas.**

6. Calificación

- Gablema (30 puntos)
- Pantema (30 puntos)
- GabPantema (40 puntos)
- Reglas de entrega (MAX -30 puntos)
- Falta de comentarios (-5 puntos c/u)
- Falta de README (-20 puntos)