

PLP - Segundo Parcial - 1^{er} cuatrimestre de 2024

#Orden	Nro. Libreta	Apellido y Nombre	Ej1	Ej2	Ej3	Nota Final

Este examen se aprueba obteniendo al menos dos ejercicios bien menos (B-) y uno regular (R). Las notas para cada ejercicio son: -, I, R, B-, B. Entregar cada ejercicio en hojas separadas. Poner nombre, apellido y número de orden en todas las hojas, y numerarlas. Se puede utilizar todo lo definido en las prácticas y todo lo que se dio en clase, colocando referencias claras. El orden de los ejercicios es arbitrario. Recomendamos leer el parcial completo antes de empezar a resolverlo.

Ejercicio 1 - Resolución

a) Representar en forma clausal las siguientes fórmulas de lógica de primer orden:

- $\forall X.(\text{Radio}(X) \implies \text{Ruidoso}(X))$
Todas las radios son ruidosas.
- $\forall X.\forall Y.((\text{Posee}(X, Y) \wedge \text{Libro}(Y)) \implies \neg \exists Z.(\text{Posee}(X, Z) \wedge \text{Revista}(Z)))$
Toda persona que posee un libro no posee revistas.
- $\forall X.(\text{EsDormilon}(X) \implies \neg \exists Y.(\text{Posee}(X, Y) \wedge \text{Ruidoso}(Y)))$
Las personas que son dormilonas no poseen nada ruidoso.
- $\exists X.(\text{Posee}(\text{pepe}, X) \wedge (\text{Libro}(X) \vee \text{Radio}(X)))$
Pepe posee un libro o una radio.

b) Utilizando resolución, determinar si la siguiente fórmula es consecuencia del conjunto anterior:

$\text{EsDormilon}(\text{pepe}) \implies \neg \exists Z.(\text{Posee}(\text{pepe}, Z) \wedge \text{Revista}(Z))$
Si Pepe es dormilón, entonces no posee revistas.

Indicar la sustitución utilizada en cada paso. Es importante tener un plan, ya sea escrito o en la cabeza.

c) Mirando la resolución utilizada en el punto anterior, ¿fue SLD? Justificar.

Ejercicio 2 - Programación Lógica

Implementar los predicados respetando en cada caso la instanciación pedida. Los generadores deben cubrir todas las instancias válidas de aquello que generan sin repetir dos veces la misma. No usar cut (!) ni predicados de alto orden como **setof**, con la única excepción de **not**.

En este ejercicio ayudaremos a un grupo de jurados a analizar las notas de las materias que cursaron algunos estudiantes para presentarse a concurso. Se cuenta con el predicado **estudiante(?E)** que es verdadero cuando **E** es un estudiante, y con el predicado **notas(-XS)** que instancia en **XS** una lista de triplas correspondientes a las notas de los estudiantes en cada materia de la siguiente forma: **(Estudiante, Materia, Nota)**. En la lista de notas pueden aparecer aplazos, pero se asume que está bien formada (es decir, puede haber muchos aplazos para un estudiante en una materia, pero a lo sumo un aprobado). Se pide:

a) Definir el predicado **tieneMateriaAprobada(+E,+M)** que es verdadero cuando el estudiante **E** tiene la materia **M** aprobada (es decir, la nota es mayor o igual a 4).

b) Definir el predicado **eliminarAplazos(+NS,-L)** que es verdadero cuando **NS** es una lista de notas y **L** es la misma lista, pero eliminando los aplazos. Los aplazos sólo pueden eliminarse si el estudiante finalmente aprobó la materia. Por ejemplo:

```
?- eliminarAplazos([(juan,plp,3),(juan,plp,9),(maria,tlen,2)],L).
L = [(juan,plp,9),(maria,tlen,2)].
```

c) Definir el predicado **promedio(+A,-P)** que es verdadero cuando **A** es un estudiante, y **P** es el promedio de todas sus notas luego de eliminar los aplazos.

Sugerencia: armar una lista de notas de **A**.

d) Definir el predicado **mejorEstudiante(-A)** que es verdadero cuando **A** es el estudiante cuyo promedio es el más alto. Puede haber más de una solución en caso de que haya más de un estudiante con el promedio más alto. En este inciso no está permitido utilizar estructuras auxiliares.

Ejercicio 3 - Objetos y Deducción Natural

a) I. Considerar las siguientes definiciones:

```
Object subclass: A [  
  a: x b: y  
    ^ x a: (y c) b: self.  
  
  c  
    ^ 2.  
]  
  
A subclass: B [  
  a: x b: y  
    ^ y c + x value.  
  
  c  
    ^ 1.  
]  
  
B subclass: C [  
  a: x b: y  
    ^ x.  
  
  c  
    ^ [self a: super c b: self].  
]
```

Hacer una tabla donde se indique, en orden, cada mensaje se envía, qué objeto lo recibe, con qué colaboradores, en qué clase está el método respectivo, y cuál es el resultado final de cada colaboración tras ejecutar el siguiente código:

```
(A new) a: (B new) b: (C new)
```

II. Implementar un método para el mensaje `#divisores`, cuyo objeto receptor es un número entero, que devuelve una colección con sus divisores.

Sugerencia: utilizar el mensaje binario `#'\'` que devuelve el resto de la división entera entre dos números.

b) Demostrar en deducción natural que vale la siguiente fórmula sin usar principios de razonamiento clásicos:

$$\forall X.\forall Y.(\exists Z.(P(X,Z) \wedge P(Z,Y)) \implies \exists W.P(X,W))$$