

PLP - Segundo Parcial - 1^{er} cuatrimestre de 2023

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 - Programación Lógica

- a) Sea un alfabeto un conjunto de átomos cualesquiera, y sea una palabra en cierto alfabeto A una lista finita de átomos que pertenecen a A. Implementar el predicado palabra(+A, +N, -P), que genera las palabras en A cuya longitud es de exactamente N elementos. Tanto en este inciso como en el siguiente, no es obligatorio respetar el mismo orden de los ejemplos.

```
?- palabra([a,b,c], 4, P).  
P = [a,a,a,a] ;  
P = [a,a,a,b] ;  
...  
P = [b,c,a,b] ;  
...  
P = [c,c,c,c] ;  
false
```

- b) Sea una frase en cierto alfabeto A una lista finita de palabras no vacías en A. Implementar el predicado frase(+A, -F), que genera todas las posibles frases en A.

```
?- frase([q,u,e,s,o], F).  
F = [] ;  
...  
F = [[q,u,e],[e,s],[e,s,o]] ;  
...  
F = [[q,u,e,s,o,s]] ;  
...
```

- c) Indicar si el parámetro P del predicado palabra y el parámetro F del predicado frase son reversibles. Justificar la respuesta.

Ejercicio 2 - Resolución

- a) Representar en forma clausal las siguientes fórmulas de lógica de primer orden referidas a números naturales:
- $\forall X \forall Y (X = Y) \supset divide(X, Y) \wedge divide(Y, X)$ - Si dos números son iguales, se dividen mutuamente.
 - $\forall X \forall Y \forall Z (divide(X, Y) \wedge divide(Y, Z)) \supset divide(X, Z)$ - La relación de divisibilidad es transitiva.
- b) Usando el método de resolución demostrar la siguiente fórmula:
 $\forall X \forall Y (X = Y) \supset divide(X, X)$
- c) La resolución realizada en el punto anterior, ¿fue SLD? Justificar. Si no lo fue, ¿sería posible encontrar una resolución SLD para este conjunto de cláusulas? (No es necesario escribirla, solamente indicar por qué se puede saber que es posible o que no lo es.)

Ejercicio 3 - Objetos

a) Definir en Cálculo ς la clase *Pokemon*, cuyas instancias puedan responder a los mensajes *salud*, *atacar* y *evolucionar*. La salud inicial de los pokemons es 100, al atacar reducen en 1 la salud de su oponente y al evolucionar, su ataque se incrementa en 1, es decir, cuando un pokemon evoluciona, hace 1 más de daño a su oponente que antes de evolucionar. Tener en cuenta que los pokemons pueden evolucionar más de una vez.

b) Se definen los siguientes objetos:

```
def [ cantidadDePokemons  $\doteq$  cero,  
    adoptar =  $\varsigma(z)\lambda(x)z.cantidadDePokemons := z.cantidadDePokemons.succ$  ]  
pikachu  $\stackrel{\text{def}}{=} Pokemon.new$ 
```

cero es el objeto definido en clase, y se tiene que $cero.succ \longrightarrow uno'$. Mostrar cómo reduce $ash.adoptar(pikachu)$. Está permitido usar la regla APP vista en clase.