

Ejercicios de repaso Final

Definiciones recursivas y demostración por recursividad

1. Para cada una de las siguientes expresiones lógicas indicar:

- Son expresiones SL o no?
- En caso de ser expresiones SL dar su árbol sintáctico.

- a. $\neg(P \wedge (Q \vee \neg R))$
- b. $(P \vee Q) \vee (\neg R \wedge P)$
- c. $((P \vee Q) \wedge (R \vee \neg(Q)))$
- d. $(\neg(R \vee (P \wedge \neg Q)) \wedge (R \vee P))$

2. La notación polaca inversa (RPN - Reverse Polish Notation) es una notación aritmética utilizada en algunas calculadoras y también se utiliza para implementar la evaluación de expresiones en un compilador. La notación RPN es una notación postfija, en la que el operador se escribe después de los operandos. Por ejemplo la expresión

$$- (3+4)/(2*5+6)$$

se escribe así:

$$3\ 4\ +\ 2\ 5\ *\ 6\ +\ /\ -$$

- Dar una definición recursiva de las expresiones aritméticas RPN.
- Escribir la expresión $(-2*5+3/(4-9)/7*-1)$ en notación RPN y dar su árbol sintáctico.

3. Se tienen las siguientes relaciones de parentesco:

padre(p, r). % p es padre de r

padre(s, w).

padre(r, x).

padre(s, q).

padre(p, s).

padre(x, y).

- Utilizando la representación gráfica de árboles, ilustrar las relaciones indicadas.
- Usando la definición recursiva de árbol binario, indicar la representación textual de este árbol.

4. Indicar el resultado de las siguientes unificaciones con listas:

- $[1,2,3,4] = [X|Y]$
- $[1,2,3,4] = [X,Y]$
- $[X|[b,c,a]] = [1|Y]$
- $[Z] = [1]$

5. Dos listas son iguales si coinciden en todos los elementos (orden y valor de cada elemento).

- a) Dar una definición recursiva de la igualdad de listas.
- b) Opcional: Escribir un procedimiento recursivo que verifique si dos listas son iguales.

6. Demostrar las siguientes propiedades de los árboles binarios:

- Todo árbol binario de altura h tiene como mínimo $h+1$ nodos.
- Demostrar que todo árbol binario de n nodos tiene como máximo altura $n-1$.
- Todo árbol **ternario** (cada nodo puede tener hasta 3 hijos) de altura h tiene como máximo $(3^{h+1}-1)/2$ nodos.
- Cuántas hojas tiene como máximo un árbol de altura h ?

7. Expresiones duales

- Demostrar que la negación del dual de una expresión lógica simple (SL) es equivalente a la expresión original reemplazando todos los literales por su negación.

Aplicaciones de la Recursividad a la programación

1. Se tiene la siguiente secuencia de sentencias. Partiendo del estado inicial vacío, indicar el estado final:

```
x = 1
y = x**2
z = x+y
```

2. El siguiente es el programa para calcular el n-ésimo número de Fibonacci:

```
% fib/2 : Calcula el n-esimo numero de Fibonacci
% arg1 : Posicion n
% arg2 : Numero de Fibonacci Fn
fib(0,0) :- !.
fib(1,1) :- !.
fib(N,Fn) :- M1 is N-1, fib(M1,P), M2 is N-2, fib(M2,Q), Fn is P+Q.
```

- Indicar las precondiciones para este programa.
- Indicar las postcondiciones para este programa.
- Determinar si bajo estas precondiciones y postcondiciones el programa es parcialmente correcto o totalmente correcto.

3. El siguiente programa determina si un número es primo:

```
% esPrimo/1 : Determina si N es un numero primo
esPrimo(N) :- M is N-1, noDivisible(N,M).

% noDivisible/2 : Determina si N no es divisible por algun natural
% entre M y 2.
% arg1 : Natural N
% arg2 : Contador M desde N hasta 1.
noDivisible(N,_) :- N<2, !, false. % 0 y 1 no son primos
```

```
noDivisible(_,1) :- !. % Terminar cuando el contador llegue a 1
noDivisible(N,M) :- N mod M =\= 0, J is M-1, noDivisible(N,J).
```

(Recordar que $N \bmod M$ calcula el residuo de la división)

- d) Indicar las precondiciones para este programa.
- e) Indicar las postcondiciones para este programa.
- f) Determinar si bajo estas precondiciones y postcondiciones el programa es parcialmente correcto o totalmente correcto.