

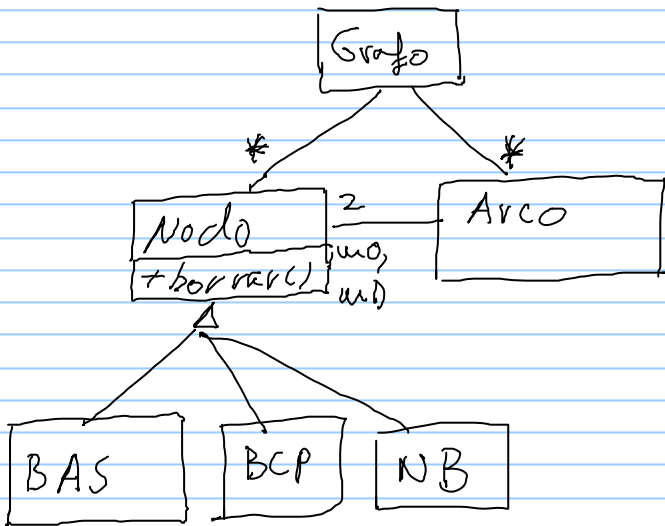
# EJERCICIOS DE T.D.D.

Título de la nota

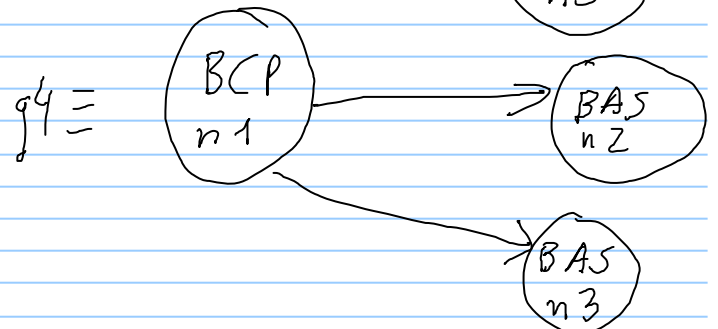
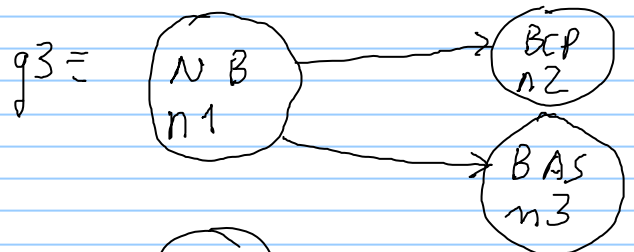
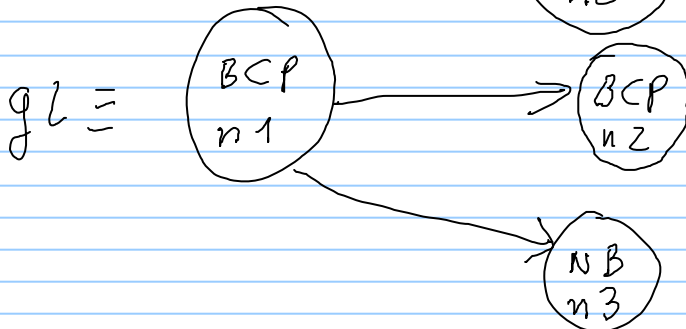
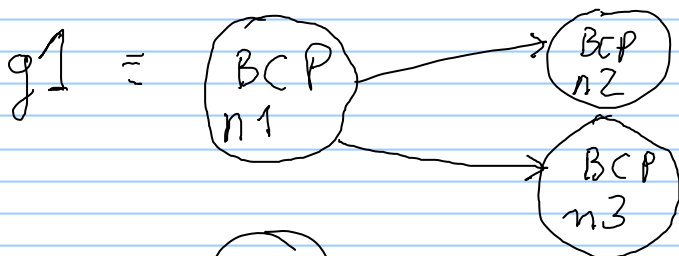
26/04/2005

## Ejercicio 1.

Se dispone del siguiente DC, q. representa grafos con nodos borrables a secas, borrables con propagación y no borrables (véase ejercicio 3 del documento "Semafros 3.pdf").



Se dispone de una clase Unit con varias "fixtures", que crean los siguientes grafos:



Se pide:

Escriba casos de prueba que, utilizando esas "fixtures", comprueben el correcto funcionamiento de las clases Nodo, Arco, Grafo, BCP, BAS y NB.

Nota: Si lo necesita, añada las operaciones que considere oportunas a las clases citadas.

## Ejercicio 2.

Se desea construir una herramienta que, tomando un fichero Java compilado (.class), sea capaz de generar casos de prueba para esa clase.

Pregunta 1. ¿Es posible? En caso afirmativo, ¿cómo? En caso negativo: ¿por qué no?

Suponiendo en cualquier caso que se pueda, el deseo es que la herramienta

mientras permita generar los casos de prueba a partir de dos mecanismos:

(a) Especificando las secuencias de métodos mediante una expresión regular (p.ej.) para una cuenta bancaria:  $\text{ingresar}(\text{float}).[\text{retirar}(-)|\text{ingresar}(-)]^*$

(b) Usando la máquina de estados de la clase para generar las secuencias de operaciones.

### Preguntas 2 y 3.

Para el caso (a), ¿qué herramienta para analizar las expresiones regulares?

Para el caso (b), y suponiendo que seamos capaces de importar máquinas de estados de alguna herramienta externa, ¿qué herramienta para manipular las máquinas de estados y para obtener de ellas casos de prueba?

En cualquiera de los casos, ¿qué  
había para bregar con parámetros  
de tipos complejos (p.ej; para la operación  
transferir (Cuenta, float))?

Los casos de prueba llevan una o más  
instrucciones tipo "assert". ¿Señala capaz  
de obtener de manera automática estas  
aserciones?

¿Considera que una herramienta de  
este tipo es una herramienta MDA?

¿Por qué?