

Práctica #6: Exhaustive Testing

1) Escribir una gramática Libre de contexto para que construya todos los términos formados por las operaciones de la clase Seq:

```
class Seq {  
    Seq()  
    Seq add();  
    Seq clear();  
    Seq concat(Seq s1, Seq s2);  
}
```

2) Dada la siguiente estructura:

```
class Node {  
    Node child0;  
    Node child1;  
}  
class BinaryDiGraph {  
    Node root;  
}
```

y el siguiente invariante de representación:

```
boolean repOK() {  
    return true;  
}
```

¿Cuántas instancias se producirán con 2 instancias de Node y 1 instancia de BinaryDiGraph?

3) Escribir el repOK() para que la siguiente estructura sea una lista doblemente encadenada :

```
class Node {  
    Node next;  
    Node previous;  
}  
class DLList {  
    Node first;  
    Node last;  
    int size;  
}
```

