

JML - TD 2

Yves LEDRU
septembre 2013

Les fichiers décrits dans ce TD sont déposés sur le site Moodle du cours:
<http://imag-moodle.e.ujf-grenoble.fr/course/view.php?id=93>

1 Implantation d'un ensemble par un arbre

On considère la structure de d'arbre définie lors du premier TD. Cette classe est complétée par des préconditions sur `min` et `max`, des postconditions sur `insert` et `delete`, ainsi que la définition de la méthode `contains`.

```
1  /*@ nullable_by_default @*/
   public class SetAsTree{
       public Integer val;
       public SetAsTree ltree;
5      public SetAsTree rtree;

       //@ public invariant ((val != null) || (ltree == null && rtree == null));
       //@ public invariant ((ltree == null) || (!ltree.emptySet() && ( ltree.max() < val.intValue() )));
       //@ public invariant ((rtree == null) || (!rtree.emptySet() && ( rtree.min() > val.intValue() )));
10      //@ public invariant (* no cycle in the tree *);

       // Constructor
       public SetAsTree(){ // produces an empty set
           val = null;
15          ltree = null;
           rtree = null;
       }
       public SetAsTree(int v){ // produces a singleton node
           val = new Integer(v);
20          ltree = null;
           rtree = null;
       }
       public SetAsTree(int v, SetAsTree l, SetAsTree r){ // arbitrary node
           val = new Integer(v);
25          ltree = l;
           rtree = r;
       }

       //getters and setters
       public Integer getVal() {
           return val;
30      }
       public void setVal(Integer val) {
           this.val = val;
35      }
       public SetAsTree getLtree() {
           return ltree;
       }
       public void setLtree(SetAsTree ltree) {
40          this.ltree = ltree;
       }
       public SetAsTree getRtree() {
           return rtree;
       }
45      public void setRtree(SetAsTree rtree) {
```

```

        this.rtree = rtree;
    }

    // Application specific methods
50
    //@ ensures contains(v);
    public void insert(int v){
    }

55
    //@ ensures !contains(v);
    public void delete(int v){
    }

    // Pure functions used in the specification
60
    public /*@ pure @*/ boolean emptySet(){
        return val == null;
    }

    //@ requires !emptySet();
65
    public /*@ pure @*/ int min(){
        if (ltree != null && ltree.getVal().intValue() < val.intValue()){return ltree.min();}
        else return val.intValue();
    }

    //@ requires !emptySet();
70
    public /*@ pure @*/ int max(){
        if (rtree != null && rtree.getVal().intValue() > val.intValue()){return rtree.max();}
        else return val.intValue();
    }

75
    public /*@ pure @*/ boolean contains(int v){
        if (val == null) {return false;}
        else if (v == val.intValue()) {return true;}
        else if (v > val.intValue() && (rtree!= null)) {return rtree.contains(v);}
        else if (v < val.intValue() && (ltree!= null)) {return ltree.contains(v);}
80
        else {return false;}
    }

    // Non side-effecting methods
85
    public /*@ non_null @*/ String toString(){
        String s = "";
        if (ltree != null) {s=s+ltree.toString();}
        s = s+" "+val+" ";
        if (rtree != null) {s=s+rtree.toString();}
        return s;
90
    }
    public void skip(){ } // useful to test the invariant.
}

```

2 Question 1

Insérez l'implantation des deux opérations `insert` et `delete`, réalisée lors du premier TD et vérifiez que cette implantation satisfait bien les postconditions de ces deux opérations.

3 Question 2

1. Complétez l'invariant pour que l'arbre soit équilibré, c'est-à-dire que la profondeur de l'arbre de gauche soit égale à celle de l'arbre de droite, à une unité près.
2. Ecrivez des tests pour JUnit qui mettent en évidence que votre implantation de `insert` et `delete` ne satisfait pas ce nouvel invariant.

4 Question 3

Donnez l'implantation **la plus simple possible** des deux opérations `insert` et `delete`, afin que la post-condition et l'invariant soient respectés. Testez votre implantation avec les fichiers mis à votre disposition et ceux que vous avez réalisés pour la question 2.

Complétez la spécification en exprimant en français les propriétés attendues de ces opérations.