```
@@ -465,36 +467,65 @@ public Side side(final Hyperplane<S> hyperplane) {
465 467      * transform to the instance
466 468      */
467 469     public AbstractRegion<S, T> applyTransform(final Transform<S, T> transform) {
468     -       return buildNew(recurseTransform(getTree(false), transform));
    470 +
    471 +       // transform the tree, except for boundary attribute splitters
    472 +       final Map<BSPTree<S>, BSPTree<S>> map = new HashMap<BSPTree<S>, BSPTree<S>>();
    473 +       final BSPTree<S> transformedTree = recurseTransform(getTree(false), transform, map);
    474 +
    475 +       // set up the boundary attributes splitters
    476 +       for (final Map.Entry<BSPTree<S>, BSPTree<S>> entry : map.entrySet()) {
    477 +           if (entry.getKey().getCut() != null) {
    478 +               @SuppressWarnings("unchecked")
    479 +               BoundaryAttribute<S> original = (BoundaryAttribute<S>) entry.getKey().getAttribute();
    480 +               if (original != null) {
    481 +                   @SuppressWarnings("unchecked")
    482 +                   BoundaryAttribute<S> transformed = (BoundaryAttribute<S>) entry.getValue().getAttribute();
    483 +                   for (final BSPTree<S> splitter : original.getSplitters()) {
    484 +                       transformed.getSplitters().add(map.get(splitter));
    485 +                   }
    486 +               }
    487 +           }
    488 +       }
    489 +
    490 +       return buildNew(transformedTree);
    491 +
469 492     }
470 493
471 494     /** Recursively transform an inside/outside BSP-tree.
472 495      * @param node current BSP tree node
473 496      * @param transform transform to apply
```