

```

package player;

import edu.neu.ccs.demeterf.demfgen.lib.List;
import edu.neu.ccs.evergreen.ir.Relation;
import gen.*;

/**
 * This class handles Shannon cofactor computations.
 */
public class Shannon {

    /**
     * Compute the Shannon cofactor for the given raw material and literal.
     *
     * @param rm
     *     The raw material.
     * @param l
     *     The literal.
     * @return The raw material representing the Shannon cofactor.
     */
    public static RawMaterial cofactor(RawMaterial rm, Literal l) {
        RawMaterial result;

        Variable litVar = l.var;

        List<Constraint> reducedConstraints = List.create();

        for (Constraint c : rm.instance.cs) {
            List<Variable> vars = c.vs;

            if (vars.contains(litVar)) {
                // Constraint fields needed for reconstruction
                Weight weight = c.w;
                int relNum = c.r.v;

                // Parameters needed for Relation.reduce()
                int varPos = vars.index(litVar);
                int sign = l.value.sign();

                // Reduce the relation based on the literal's sign
                relNum = new Relation(vars.length(), relNum).reduce(varPos,
                    sign);

                reducedConstraints = reducedConstraints.append(new Constraint(
                    weight, new RelationNr(relNum), vars));
            }
        }

        // Never return an empty raw material
        if (reducedConstraints.isEmpty()) {
            result = rm;
        } else {
            result = new RawMaterial(
                new RawMaterialInstance(reducedConstraints));
        }

        return result;
    }
}

```