

```

package player.satsolver;

import java.util.HashMap;
import java.util.HashSet;
import java.util.Map;
import java.util.Set;

import edu.neu.ccs.satsolver.InputInitialI;
import edu.neu.ccs.satsolver.PairI;
import gen.Type;
import gen.TypeInstance;

public class InputInitial implements InputInitialI {

    private final Set<PairI> pairs;

    public InputInitial(Type t) {
        Map<Integer, Integer> types = new HashMap<Integer, Integer>();
        int total = 0;
        for (TypeInstance i : t.instances) {
            Integer n = types.get(i.r.v);
            if (n == null) {
                types.put(i.r.v, 1);
            } else {
                types.put(i.r.v, n + 1);
            }
            ++total;
        }
        pairs = new HashSet<PairI>();
        for (Map.Entry<Integer, Integer> e : types.entrySet()) {
            pairs.add(new Pair(e.getKey(),
                ((double) e.getValue()) / ((double) total)));
        }
    }

    public Set<PairI> getPairs() {
        return pairs;
    }
}

```