DPJizer: A Tool for Automated Java to DPJ Transformation

by

Mohsen Vakilian, et al.

Automated Software Engineering 2009,

November 16-20, 2009 in Auckland, New Zealand

What is DPJ?

DPJ is Deterministic Parallel Java. In today's widely used parallel programming models, subtle programming errors can lead to unintended nondeterministic behavior and hard to catch bugs. In contrast, a parallel programming model that is deterministic by default: deterministic behavior is guaranteed unless the programmer explicitly uses nondeterministic constructs.

How effect system in Java helps?

Effect systems are important for reasoning about the side effects of a program. Effect Systems helps in machine checked documentation Effect Systems enable composable easily & correctly

What is a region?
Region is a collection of locations

What is an effect?
Effect is a read or a write operation on a region.

Effect annotations describing the side effects of each method can enable modular analysis of effects. But, although these annotations have been around for decades, they have not been used much in practice. The reason is that manually writing such effects is tedious and error-prone. Moreover when codes change, the annotations needs to be changed too. Thus DPJizer (essentially an Eclipse plug-in) will help in automatically inferring, the effects of each program statement. Moreover it has the ability to infer effects for programs even on nested heap regions, including recursive as well as non-recursive data structures.
So, if we have a code like :-

class Node<region P> {
    region L,R;
    double mass in P;
    Node <P:L> left in P:L;
    Node <P:R> right in P:R;

    void setMassforTree(double mass)
    {
        this.mass = mass ; //*1
        cobegin{
            if(left != NULL) {left.setMassforTree(mass);} //*2
            if(right != NULL) {right.setMassforTree(mass);} //*3
        }
    }
}

Summary

DPJizer is an interactive tool for inferring effect annotations given the region annotations

DPJizer infers effect that are sufficiently precise and fine-grained.

References :-

[1] UPCRC Seminar on 2009-11-05