

## **Wikiprint Book**

**Title:** projects/jpf-mango/archive/FourthExample

**Subject:** Java Path Finder - projects/jpf-mango/archive/FourthExample

**Version:** 1

**Date:** 02/21/2013 01:56:18 PM

## Table of Contents

Example 4: Local Forward Flow	3
Local forward flow of ItsAWrap.clear() loop	3

#### Example 4: Local Forward Flow

To help the user [trade in conjectures for hypotheses?](#) Mango determines the *local forward flow* of a loop. This is the execution flow from the entry point of the loop to each edge which leaves the loop. These exits may be classified as *good* or *bad*. In the graph below, the forward flow edges are red. The good exit is the return, and the two bad exits are exceptions which may be thrown during the attempt to make the assignment  $x[i]=0$ . Since  $x$ , viewed as a heap reference, is a loop invariant, the null pointer exception already generates the hypothesis  $x$  is *undefined*. However, the array index out of bounds exception gives rise to the conjecture  $i^{\wedge}$  is greater than or equal to length of the array  $x$  OR  $i^{\wedge}$  is less than 0 which does not automatically simplify because  $i$  is not a loop invariant. The conjecture to leave the loop in the good case is  $i^{\wedge}$  is greater than or equal 10. By traversing the forward flow, Mango can generate all this information and present it in an orderly fashion. The user may then deduce the hypothesis which implies the conjecture to leave the loop:  $x$  is defined and the length of  $x$  is greater than or equal to 10. The challenge is to automate this procedure with a more sophisticated reasoning process. What level of generality to address, and what strategy to implement, are interesting questions to ask.

#### Local forward flow of `ItsAWrap.clear()` loop

