

Output-dependence

Where a register is written and then written to again (the second write must occur after the first following any transformation).

Anti-dependence

Where a register is read and then written to (the write must occur after the read following any transformation).

Knowing a register is written once means output-dependence and anti-dependence don't occur. This property means that our previous local optimizations can be applied globally. To handle loops and other branches, special phi instructions encode the merge of particular values from a different place on the control-flow graph.

Array SSA form is an extension to SSA form in which loads and stores are considered to be defining a special variable called a heap. Modeling memory accesses in this way allows the compiler to reason that if two reads occur from the same array location with a heap with the same name, the second read can be replaced with a copy of the first. Redundant stores are handled in a similar way. It also allows accesses to nonrelated heaps to be reorganized—for example, with floating-point and integer operations. Array SSA form was originally devised for FORTRAN; extended array SSA form adds to the form factors, such as Java's fields being unable to alias with one another (Fink et al. 2000).

Also in the SSA form, Jikes RVM constructs pi instructions, placed after a branch uses an operand. The pi instruction uses the same operand as the branch and gives it a new name to be used in place of the operand in subsequent instructions. Using pi instructions, the compiler can reason that a branch performs a test, such as a null test, and that any null tests using the register result of the pi instruction are redundant. Similarly, array bound checks can be removed (Bodik et al. 2000).

The loop versioning optimization in the HIR SSA form also removes possible exceptions. Loop versioning moves exception-checking code out of loops and explicitly tests whether the exceptions can occur before the loop is entered. If an exception can occur, then a version of the loop with exception-generating code is executed. If an exception cannot occur, then a loop version without exceptions is executed.

Partial evaluation

The HIR optimizations, including the SSA optimizations, are able to reduce the complexity of a region of code, but this can be limited by what constant values are available. Often a value cannot be determined to be constant when it comes from an array, which in Java can always have their values altered. The most common occurrence of this is with strings. Jikes RVM introduces the pure annotation as an extension to Java that enables a method with constant arguments to be evaluated, using reflection, at compile time. Leveraging annotations and reflection for this purpose is straightforward in a metacircular runtime.