1 How do the declarations look in the GIMPLE? Specifically, what happens to multiple variables declared and initialized on the same line? What happens to global declaration [int Z]?

Ans Each declaration is displayed on its own line as an independent local variable declaration. Initializers are broken up into simple declarations at the top and an assignment statement in the body of the procedure. There is no declaration for global variables; they

are used directly.

2 How are complex expressions such as [c = a \* b + 25] broken down?

Can you determine a rule as to when temporary variables will be introduced?

Ans A GIMPLE statement can have at most one operation. Any expression containing more than one operation is broken up by introducing temporaries. It is easy to convert C expressions to GIMPLE statements by considering abstract syntax trees, where each non-leaf node must correspond to either a local or temporary variable storage.

3 How are floats/doubles represented in GIMPLE? What happens when a float/double is assigned to an integer in [Z = r]?

Ans Floats/doubles are displayed in scientific notation, and any

implicit assignments to integers are converted to explicit casts in

GIMPLE.

4 How are reads/writes to the global variable Z performed? Why is there a temporary introduced for the statement [Z = Z + 1] but not

for the statements [q = Z] or [Z = p]?

Ans Reading/writing from/to global variables requires a memory load or store operation. As GIMPLE statements can have at most of operation, the only type of statements involving globals are of the form "GLOBAL = LOCAL" or "LOCAL = GLOBAL". This is because while local variables of a procedure can reside entirely in registers, every read/write to global variables must refer to main memory as they can be accessed by other entities (threads, I/O devices, etc). Hence, any operation on global variables (such as the increment shown in this example) has to go through an indirection of a temporary version of the global inside registers (as this temporary value is not visible to I/O devices, etc). Global temporaries are named similar to the global variable but with a numeric suffix.