

Why loop variants are integers

 archive.eiffel.com/doc/faq/variant.html

This site contains older material on Eiffel. For the main Eiffel page, see <http://www.eiffel.com>.

The Eiffel Support Pages

[This page reproduces a message posted on comp.lang.eiffel on 27 July 1995. The message was prepared jointly by Michael Schweitzer, then of Swissoft (Goettingen, Germany), and Bertrand Meyer of ISE.]

A loop's variant is an expression that serves to convince software developers, the readers of their software, and proof tools if available, that a loop will terminate.

Mathematically, a loop variant should for full generality take its values in any set that is "well-founded". A set equipped with an order relation " $<$ " is called well-founded (this is also known as the Descending Chain Condition or DCC) if any decreasing sequence is stationary, based on the following definitions:

- A **sequence** is a total function from \mathbb{N} (the set of natural integers) to S .
- A sequence s is **decreasing** if, for any integer i , $s(i+1) \leq s(i)$ where \leq denotes the order relation on S . (Note the use of " \leq ", not " $<$ ": "decreasing" here does not necessarily mean "strictly decreasing"; two successive elements of a decreasing sequence may be equal.)
- A sequence s is **stationary** if there exists some integer n such that $s(i) = s(n)$ for all $i \geq n$.

An example of a well-founded set is \mathbb{N} itself with the usual order relation: any decreasing sequence, such as 25, 21, 21, 12, 6, ... cannot go on strictly decreasing forever, so it has to be stationary - e.g. it can go down to 0 and stay there forever.

As another example, involving a partial order rather than a total one, consider a tree, finite or infinite, where the relation " \leq " between nodes is defined by: $a \leq b$ if node a is an ancestor of node b (that is to say, a is either b or, recursively, the parent of an ancestor of b). Then every decreasing sequence is stationary: starting from any node, you will either stay forever at some intermediate ancestor or, sooner or later, end up at the root.

On the other hand, the set of non-negative real (or rational) numbers with the usual " \leq " is not well-founded; see e.g. the sequence 1, $1/2$, $1/3$, $1/4$, ..., i.e. $s(i) = 1/(i+1)$, which is decreasing but not stationary.

Now for the application to software and to the Eiffel loop structure. Assume that we have

a loop and we can associate with it an expression ev (the variant expression) such that:

- 1 - The value of ev at any time is an element of a set V that is well-founded for a certain relation \leq .
- 2 - Every execution of the loop body replaces the value bv that ev has before the loop by a new value av ('a' for 'after') such that $av < bv$ (that is to say, $av \leq bv$ and $av \neq bv$).

Then the loop, by the very definition of "well-founded", will terminate - i.e. will have a finite number of iterations.

[Note: for anyone who wants to probe further there are many good introductory books on the theoretical basis for programming.]

One could of course define a class WELL_FOUNDED (which, by the way, would have no connection whatsoever with NUMERIC, as the tree example shows). But from a software engineering perspective that seems overkill. Assume someone gives you a loop backed by an argument claiming that the loop will terminate, based on a variant that takes its value in some strange set. It would be natural for you to reply:

"Very impressive. But could you please you give me an estimate - no, actually just an upper bound will do - on the number of iterations there will be, depending of course on the state in which the loop is started?"

If the loop author gives you an answer (a correct one), then that answer is an alternative variant - an integer variant. Remember that the answer doesn't have to be precise; it just has to be an upper bound (e.g. something on the order of 2^n , where n is the real number of iterations, is OK as a variant - although in practice you will probably not be happy as a performance-conscious software engineer until you have a better estimate!).

But assume the loop author is unable to answer the above request. You probably will think that there is something fishy. Few people would trust a loop for which termination is claimed but no one is able to give an integer upper bound on the number of iterations.

Thus variants are, and should remain, integers (see ETL pp. 130-131).

Michael Schweitzer

Bertrand Meyer

ISE Inc., Santa Barbara
805-685-1006, fax 805-685-6869
<http://www.eiffel.com>

