An *evaluation* is the execution of a statement, expression, constructor, destructor, initialization of an object (including unnamed temporary objects).

Two evaluations, Xi and Xj of the same statement, expression, etc., X in two different vector applications i and j of F might be *horizontally matched*, according to the following rules.

* If X is the top-level statement in F, then Xi is horizontally matched with Xj.
* If Y is an unconditionally-executed sub-part of X and Xi is horizontally matched with Xj then Yi is horizontally matched with Yj.
* If Xi is horizontally matched with Xj and X is a selection statement, conditional expression, or an evaluation of (built-in) operator && or operator ||, and both evaluations select the same sub-statement or sub-expression Y, then Yi is horizontally matched with Yj.
* If Xi is horizontally matched with Xj and X is an iteration statement, then each evaluation of the (immediate) subexpressions and substatements of X in application i is horizontally matched with the corresponding subexpression and substatement in application j, where *corresponding* refers to the ordered sequence of iterations, regardless of the values of the loop control variables. If the application execute different numbers of iterations, the common iterations are horizontally matched and the remainder are not.
* [Some rule(s) about gotos, longjmp, throw, and unstructured switch goes here]

For distinct statements, expressions, etc. X and Y and applications i and j, where i would precede j in the sequential order of applications, *wavefront application* of F provides the following sequencing guarantees:

* If Xi is horizontally matched with Xj and Xj is sequenced before Yj, then Xi is sequenced before Yj.
* If Yi is horizontally matched with Yj and Xi is sequenced before Yi, then Xi is sequenced before Yj.